

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

«На правах рукопису»

УДК 004.94

«До захисту допущено»

Завідувач кафедри

Стіренко С.Г.

(підпис)

(ініціали, прізвище)

“ ” _____ 2020 р.

Магістерська дисертація

зі спеціальності: 123. Комп'ютерна інженерія
(код та назва напрямку підготовки або спеціальності)

Спеціалізація: 123. Комп'ютерні системи та мережі

на тему: Система доповненої реальності з функціями маркетплейсу

Виконала: студентка 6 курсу, групи Ю-91мп
(шифр групи)

Сташенко Марія Романівна

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник доцент, к.т.н., Коган А.В.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2020 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет (інститут) Інформатики та обчислювальної техніки
(повна назва)

Кафедра Обчислювальної техніки
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність 123. Комп'ютерна інженерія
(код і назва)

Спеціалізація 123. Комп'ютерні системи та мережі
(код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Стіренко С.Г.
(підпис) (ініціали, прізвище)

« » _____ 2020 р.

**ЗАВДАННЯ
на магістерську дисертацію студенту
Стащенко Марії Романівні
(прізвище, ім'я, по батькові)**

1. Тема дисертації «Система доповненої реальності з функціями маркетплейсу»

Науковий керівник дисертації к.т.н., доцент Коган А.В.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від « 26 » жовтня 2020 р. № 3132-с

2. Строк подання студентом дисертації _____

3. Об'єкт дослідження - процес синтезу доповненої реальності з елементами маркетингових стратегій.

4. Предмет дослідження - технології доповненої реальності.

5. Перелік завдань, які потрібно розробити: дослідження методів та технологій доповненої реальності, функцій маркетплейсу, оптимізація завантаження та

керування моделями, розробка програмного забезпечення для системи доповненої реальності, аналіз отриманих результатів.

6. Консультанти розділів дисертації:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
нормоконтроль	д.т.н., професор Кулаков Ю.О.		

7. Дата видачі завдання

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів дисертації	Примітка
1	Дослідження літератури та документації	15.09.2020	Викон.
2	Огляд існуючих рішень	20.09.2020	Викон.
3	Аналіз теоретичних методів	01.10.2020	Викон.
4	Проектування системи	10.10.2020	Викон.
5	Програмна реалізація системи	25.10.2020	Викон.
6	Тестування та виправлення помилок	30.10.2020	Викон.
7	Оформлення документації	03.12.2020	Викон.

Студент

(підпис)

Стащенко М. Р.

(ініціали, прізвище)

Науковий керівник дисертації

(підпис)

Коган А. В.

(ініціали, прізвище)

РЕФЕРАТ

на магістерську дисертацію

виконану на тему: Система доповненої реальності з функціями
маркетплейсу

студенткою: Стащенко Марією Романівною

Робота складається із вступу та чотирьох розділів. Загальний обсяг роботи: 82 аркуші основного тексту, 31 ілюстрації, 29 таблиць. При підготовці використовувалася література з 18 різних джерел.

Актуальність. За останні п'ять років технології доповненої реальності стали однією з найбільш обговорюваних тем в сфері програмування. Ідея створення програмного забезпечення, що розширює двомірні кордони екранів пристроїв, сама по собі є багатообіцяючою, а численні проекти з модернізації існуючих систем за допомогою AR (augmented reality - доповнена реальність) тільки підкріплюють її значимість.

У найближчі роки системи доповненої реальності повинні змінити не тільки парадигму розвитку технологій, а й погляд людини на навколишній світ. Перевага доповненої реальності полягає в тому, що існує нескінченне число варіацій її використання у всіх сферах життя. Сервіси доповненої реальності вже впроваджені в сферах освіти, медицини, реклами і, звичайно, розваг.

Очевидно, що доповнена реальність стала продуктом, який продають і купують. Найпотужніші гравці ринку намагаються монополізувати конкретні сфери і не дають шансів реалізуватися невеликим компаніям і окремим розробникам. Таким чином, створення маркетплейсу, де замовник може особисто контактувати з розробником необхідних йому для перегляду в AR моделей, здається логічним рішенням у нинішніх реаліях.

Мета роботи розроблення системи доповненої реальності, що містить функції маркетплейсу.

Завдання дослідження

- дослідження методів та технологій доповненої реальності;
- дослідження функцій маркетплейсу;
- оптимізація завантаження та керування моделями;
- розробка програмного забезпечення для системи доповненої реальності;
- аналіз отриманих результатів.

Об'єкт дослідження – процес синтезу доповненої реальності з елементами маркетингових стратегій.

Предмет дослідження – технології доповненої реальності.

Методи досліджень. Для досягнення поставлених в магістерській роботі задач, використано методи візуального моделювання, синтезу та порівняльного аналізу,.

Наукова новизна

- запропоновано більш функціонально розвинену систему доповненої реальності.
 - розроблено програмний продукт: систему доповненої реальності з функціями маркетплейсу.

Проведене дослідження дає можливість використання розробленої системи доповненої реальності без додаткових ресурсів і забезпечує повноцінну взаємодію між розробниками і клієнтами.

Особистий внесок здобувача. Магістерське дослідження є самостійно виконаною роботою, в якій відображено особистий авторський підхід та особисто отримані теоретичні та прикладні результати, що відносяться до вирішення задачі системи доповненої реальності з функціями маркетплейсу. Формулювання мети та завдань дослідження проводилось спільно з науковим керівником.

Практична цінність. Отримані результати можуть використовуватися у майбутніх дослідженнях за напрямками:

- вдосконалення систем доповненої реальності;
- синтез технологій доповненої реальності з інструментами розповсюдження продукту.

Ключові слова

Доповнена реальність, AR, маркетплейс, маркери, 3D-моделі.

ABSTRACT

for a master's thesis

made on the topic: Augmented reality system with marketplace functions

by student: Stashenko Maria Romanivna

The study consists of an introduction and four sections. Total workload is 82 sheets of body text, 31 illustrations, 29 tables. 18 various sources were used.

The urgency of the problem. Over the last five years, augmented reality technologies have become one of the most discussed topics in the field of programming. The idea of creating software that expands the two-dimensional boundaries of device screens is promising itself, and numerous projects destined to upgrade existing systems with AR (augmented reality) only reinforce its importance.

In the coming years, augmented reality systems must change not only the paradigm of technology, but also the human view of the world. The advantage of augmented reality is that there is an infinite number of variations of its usage in all spheres of life. Augmented reality services are already implemented in the fields of education, medicine, advertising and, of course, entertainment.

Obviously, augmented reality has become a product that is being sold and bought. The most powerful market players try to monopolize specific areas and do not give a chance to small companies and individual developers. Thus, the creation of a marketplace, where the customer can personally contact the developer he needs to view in AR models, seems a logical solution in today's reality.

The purpose of the study is to develop an augmented reality system that contains the functions of a marketplace.

Objectives of the study are:

- research of methods and technologies of augmented reality;
- study of marketplace functions;

- optimization of loading and management of models;
- development of software for augmented reality system;
- analysis of the obtained results.

Object of study is the process of synthesis of augmented reality with elements of marketing strategies.

Subject of study is augmented reality technology.

Research methods. To achieve the objectives set in the master's thesis, the methods of visual modeling, synthesis and comparative analysis are used.

Scientific novelty

- a more functionally developed augmented reality system is proposed.
- software product developed: augmented reality system with marketplace functions.

The study makes it possible to use the developed system of augmented reality without additional resources and provides full interaction between developers and customers.

Personal contribution of the applicant. The master's research is an independently performed work, which reflects the personal author's approach and personally obtained theoretical and applied results related to solving the problem of augmented reality system with marketplace functions. The formulation of the purpose and objectives of the study were carried out jointly with the supervisor.

Practical value. The obtained results can be used in future research in the following areas:

- improving augmented reality systems;
- synthesis of augmented reality technologies with product distribution tools.

Keywords. Augmented reality, AR, marketplace, markers, 3D models.

ЗМІСТ

ПЕРЕЛІК ТЕРМІНІВ ТА СКОРОЧЕНЬ	11
ВСТУП	12
РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	14
1.1 Доповнена реальність в освіті	14
1.1.1 Додатки для освіти з доповненою реальністю	16
1.2 Доповнена реальність у рекламі	19
1.2.1 Додатки для реклами з доповненою реальністю	20
1.3 Доповнена реальність у інших галузях	21
1.3.1 Охорона здоров'я	21
1.3.2 Подорожі та туризм	22
1.3.3 Виробництво	22
1.4 Аналіз існуючих рішень	22
ВИСНОВОК ДО РОЗДІЛУ 1	24
РОЗДІЛ 2. ПРОЕКТУВАННЯ ДОДАТКУ	26
2.1. Опис предметної області	26
2.2 Визначення вимог і завдань	28
2.3 Опис функціоналу системи	29
2.4 Прецеденти	29
2.4.1 Реєстрація	33
2.4.2 Авторизація	33
2.4.3 Перегляд моделі	33
2.4.4 Завантаження моделі	34
2.4.5 Генерація QR-коду	34
2.5 Концептуальна діаграма класів структури даних	35

2.6 Проектування графічного інтерфейсу	36
2.6.1 Мобільний додаток	36
2.6.2 Веб-додаток	38
ВИСНОВОК ДО РОЗДІЛУ 2	43
РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ ДОПОВНЕНОЇ РЕАЛЬНОСТІ З ФУНКЦІЯМИ МАРКЕТПЛЕЙСУ	44
3.1 Архітектура системи.....	44
3.2. Вибір технологій та їх обґрунтування	45
3.2.1 Технології для реалізації серверної частини.....	45
3.2.2 Технології для реалізації мобільних додатків.....	49
3.2.3. Технології для реалізації веб-застосунку	51
3.3. Реалізація WEB API.....	52
3.4. Реалізація мобільних додатків.....	55
3.5. Реалізація веб-клієнту.....	59
ВИСНОВОК ДО РОЗДІЛУ 3	65
РОЗДІЛ 4. РОЗРОБКА СТАРТАП-ПРОЕКТУ	66
4.1 Опис ідеї проекту	66
4.1.1 Задум проекту.....	67
4.1.2 Удосконалення продукту	68
4.2 Аналіз ринкових можливостей запуску стартап-проекту.....	70
4.3 Розроблення ринкової стратегії проекту	76
4.4 Розроблення маркетингової програми стартап-проекту.....	79
ВИСНОВОК ДО РОЗДІЛУ 4	85
ВИСНОВОК.....	86
Список використаної літератури	88

ДОДАТКИ.....

ПЕРЕЛІК ТЕРМІНІВ ТА СКОРОЧЕНЬ

ОС	Операційна система.
GUI	Форма користувацького інтерфейсу, яка дозволяє користувачам взаємодіяти з електронними пристроями за допомогою графічних значків і візуальних індикаторів.
AR	(від англ. Augmented reality – доповнена реальність) - система, яка виконує три основні функції: поєднання реального та віртуального світів, взаємодія в реальному часі та точна 3D-реєстрація віртуальних та реальних об'єктів.
HUD	(heads-up display) - прозорий дисплей, що представляє дані, не вимагаючи від користувачів відволікати погляд.
ARKit	Платформа розробки додатків з доповненою реальністю для IOS.
ARCore	Платформа розробки додатків з доповненою реальністю для Android.
Маркетплейс	Тип електронної комерції, де інформація про товар чи послугу надається третіми сторонами.
SLAM	(від англ. simultaneous localization and mapping) - метод одночасної локалізації та побудування карти.
RFID	(від англ. Radio Frequency IDentification) - спосіб ідентифікації об'єктів, в якому за допомогою радіосигналів зчитуються дані з RFID-міток.
Суперімпозиція	Порівняння графіків шляхом накладення їх один на одного.

ВСТУП

За останні п'ять років технології доповненої реальності стали однією з найбільш обговорюваних тем в сфері програмування. Ідея створення програмного забезпечення, що розширює двомірні кордони екранів пристроїв, сама по собі є багатообіцяючою, а численні проекти з модернізації існуючих систем за допомогою AR (augmented reality - доповнена реальність) тільки підкріплюють її значимість.

У найближчі роки системи доповненої реальності повинні змінити не тільки парадигму розвитку технологій, а й погляд людини на навколишній світ. Перевага доповненої реальності полягає в тому, що існує нескінченне число варіацій її використання у всіх сферах життя. Сервіси доповненої реальності вже впроваджені в сферах освіти, медицини, реклами і, звичайно, розваг.

Поки найбільш очевидним використанням технологій AR є багатокористувацькі ігри на основі геопозиційної доповненої реальності і інструменти для редагування фото і відео в соціальних мережах, дана робота фокусується на основних функціях таких технологій і впровадження їх у повсякденне життя.

Очевидно, що доповнена реальність стала продуктом, який продають і купують. Найпотужніші гравці ринку намагаються монополізувати конкретні сфери і не дають шансів реалізуватися невеликим компаніям і окремим розробникам. Таким чином, створення маркетплейсу, де замовник може особисто контактувати з розробником необхідних йому для перегляду в AR моделей, здається логічним рішенням у нинішніх реаліях.

Актуальність даної роботи полягає в тому, що система доповненої реальності буде включати в себе функції маркетплейсу для підвищення ефективності та продуктивності реалізації, а також залучення більшої аудиторії та розширення меж діяльності на більшу кількість сфер.

Метою роботи є розроблення системи доповненої реальності, що містить функції маркетплейсу.

Об'єкт дослідження – процес синтезу доповненої реальності з елементами маркетингових стратегій.

Предмет дослідження - технології доповненої реальності.

Також завданням роботи є створення стартап-проекту, що включатиме розробку стратегій для впровадження продукту на ринок послуг.

РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1 Доповнена реальність в освіті

Доповнена реальність в освіті має можливість впливати на поточний процес навчання. AR може змінити місце та час навчання, запровадити нові та додаткові способи та методи навчання. Можливості технології доповненої реальності роблять заняття більш привабливими, а інформацію зрозумілішою [1].

На сьогоднішній день 80% людей у віці від 18 до 25 є власниками смартфонів. Більшість з них є активними користувачами смартфонів, які використовують ці гаджети для доступу до соціальних мереж, ігор та спілкування з людьми та навколишнім світом. Тим часом набагато менша частина людей використовує телефони для навчальних цілей, для виконання домашніх завдань, пошуку інформації про предмет тощо.

Потенціал поєднання смартфонів та доповненої реальності в освіті великий, хоча і досі не є повністю розкритим. AR різними способами може надавати студентам додаткову цифрову інформацію про необхідну тему та полегшувати аналіз складної інформації. Можливість поєднання реальності та цифрового контенту безперестанно вдосконалюється, відкриваючи більше можливостей для викладачів та студентів.

Використання доповненої реальності у процесі навчання:

1) Пояснення абстрактних та складних понять

Технологія AR має можливість рендерити об'єкти, які важко уявити у просторі, і перетворювати їх у тривимірні моделі, що полегшує сприйняття абстрактного та складного вмісту об'єктів. Це особливо корисно для тих, хто навчається наочно, і практично для всіх, хто може перекласти теоретичний матеріал у реальні концепції.

2) Клас доповненої реальності

Анімована складова доповненої реальності на занятті може привернути увагу студентів набагато ефективніше, ніж традиційні візуальні інструменти, а також мотивувати їх до вивчення теми. Додавання додаткових даних, напр. біографічні факти, історичні дані про місця чи події, візуальні 3D-моделі, дають студентам ширше розуміння тем.

Виконуючи домашнє завдання, учні можуть сканувати певні елементи книги та отримувати текстові, аудіо- чи відеоповідомлення та поради від викладачів; або вони можуть знайти корисну інформацію про курс, викладача чи інших студентів, що може призвести до кращої комунікації та взаєморозуміння.

3) Залучення студентів та взаємодія

За допомогою включення доповненої реальності до занять викладачі можуть залучати учнів та студентів до процесу за допомогою тривимірних моделей [2]. Це може бути лише частиною уроку, як вступ чи передмова, або як підтримка основної теми додатковою інформацією з іншого ракурсу.

4) Моделювання об'єктів

Інтерактивне навчання, вправи для рук, тощо допомагають отримати кращі знання будь-якої дисципліни. Програми з AR для студентів-медиків можуть бути одним із способів вивчити та глибше дослідити анатомію людини. Доповнена реальність в основному означає взаємодію з 3D-моделями. Можна встановити обертання, масштаб, кольорову схему, стилі тощо. Також, можуть бути створені більш вдосконалені моделі за допомогою спеціальних пристроїв, таких як голографічні лінзи, замість смартфонів.

5) Громадські місця і установи, які мають освітню функцію

Відвідувачі музеїв можуть отримати доступ до AR через смартфони та виявити історичний вміст, пов'язаний з предметами та додаткову інформацію про те, що вони бачать. Хоча через обмеження простору чи бюджету не всі музеї та визначні пам'ятки можуть собі дозволити, як тільки AR стане

доступнішим, з'являться нові великі можливості для музеїв. Перевагою є те, що доповнена реальність вже доступна відвідувачам через мобільні пристрої.

6) Практичні заняття

У багатьох випадках теоретичних знань недостатньо для отримання належних навичок у професійних сферах. Студенти не повинні бути простими слухачами та пасивними спостерігачами. Студенти технічних факультетів особливо потребують практики та практичного досвіду у своїх галузях. Завдяки взаємодії, на відміну від VR, функції AR можуть допомогти виконати віртуальну практику - завдяки доповненим навчальним посібникам, цифровому моделюванню та моделюванню, і врешті набути певного досвіду.

1.1.1 Додатки для освіти з доповненою реальністю

1.1.1.1 Anatomy 4D

Anatomy 4D дозволяє вам детально спостерігати та вивчати анатомію людини за допомогою дивовижних технологій доповненої реальності. Додаток використовує потужність сучасних телефонів для створення зображень із доповненою реальністю, які роблять досвід навчання більш яскравим, оживляючи підручники та фотографії для кращого перегляду та перевірки.

Додаток дозволяє окремо виділяти різні системи органів, досліджувати людське тіло та його системи, миттєво перемикатися між чоловічою та жіночою анатомією та виявляти нові цільові зображення, які служать орієнтирами для системи доповненої реальності.

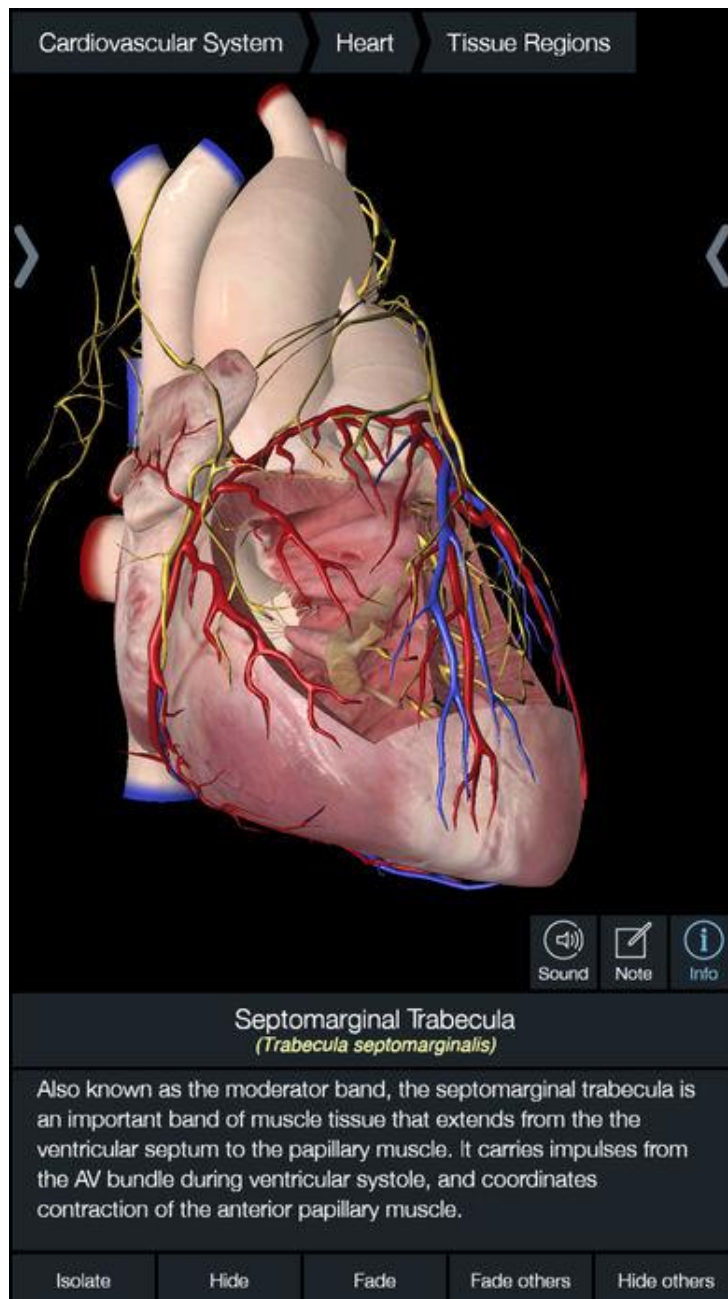


Рисунок 1.1 – Модель у додатку Anatomy 4D

Переваги:

- додаток дозволяє користувачам ізолювати різні частини людського тіла та серця;
- можливість вибору складання візуалізації органів;
- повний 360-градусний огляд об'єкта.

Недоліки:

- лише дві доступні друковані цілі;
- неможливість виділення окремих органів.

1.1.1.2 Word Lens

Word Lens використовує розпізнавання тексту, щоб визначити слово або фразу, а програмне забезпечення автоматичного перекладу перекладає його на нову мову. Потім переклад наклеюється на вихідне місце. Оптичний механізм розпізнавання символів працює із вбудованим перекладачем у режимі реального часу, щоб перекласти іноземні фрази. Потім перетворений текст відображається на екрані смартфона.

Переваги:

- корисний інструмент мовної підтримки;
- добре працює з "чітко надрукованим текстом.

Недоліки:

- слова рухаються вперед і назад у процесі перекладу;
- нерівний інтерфейс у версії Android;
- не розпізнає рукописний текст.



Рисунок 1.2 – AR у додатку Word Lens

1.2 Доповнена реальність у рекламі

Ринок цифрової реклами бурхливо розвивається: очікується, що витрати на цифрову рекламу у всьому світі зростуть з 280 мільярдів у 2018 році до 510 мільярдів доларів до 2023 року.

Майбутнє реклами належить рекламі у AR через, окрім іншого, такі фактори:

1) Зростання продажів

AR пропонує для маркетологів та рекламодавців інші способи збільшення обсягів продажів: віртуальні тести.

Натрапивши на звичайну рекламу, клієнт бачить гарні зображення товарів, але він не може бути впевненим, що вони йому пасуватимуть. Це не є проблемою, якщо використовується технологія доповненої реальності: є можливість віртуально спробувати кілька товарів і вибрати той, який найбільше подобається. Це робить AR-рекламу потужним інструментом для збільшення продажів і доходу [3].

2) Емоційний зв'язок

Реклама з доповненою реальністю є візуально привабливою, що означає, що вона допомагає маркетологам створити певний емоційний зв'язок із клієнтами. Наприклад, на відміну від зображень або банерів, AR-реклама є інтерактивною та реальною: споживачі можуть бачити та навіть взаємодіяти з нею.

Взаємодіючи з рекламою AR, споживачі відчують, що вони грають у захоплюючу відеогру. Це створює емоційний зв'язок із клієнтами, спонукаючи їх робити покупки. Емоційний зв'язок є чудовим інструментом для підвищення поінформованості про товар. Люди краще згадують бренди, з якими у них є позитивні асоціації, тому реклама AR ідеально підходить не тільки для підвищення продажів, а й для формування репутації постачальника.

3) Гіперлокальна реклама

Цифрова реклама використовує поглиблені алгоритми машинного навчання для аналізу потреб та інтересів користувачів, щоб рекомендувати правильні товари та послуги людям. Оголошення та рекламні відео з доповненою реальністю пропонують маркетологам ще кращі можливості: розширену гіперлокальну рекламу.

На відміну від традиційної реклами, яка, наприклад, повідомляє клієнтів про такі заклади, як кафе та магазини, гіперлокальна реклама AR справді захоплює, оскільки може показувати об'єкти прямо на екрані смартфона. Таким чином, клієнти не лише отримують інформацію про місця, які можуть бути для них цікавими, але й «переносяться» туди.

1.2.1 Додатки для реклами з доповненою реальністю

1.2.1.1 *IKEA Place*

Додаток доповненої реальності IKEA Place дозволяє віртуально «примірити» меблі в квартирі перед покупкою.

Версія для Android, побудована на платформі Google ARCore, яка є аналогом Apple ARKit.

За функціональністю IKEA Place для Android практично повністю ідентична версії для iOS і дозволяє вибирати меблі з широкого каталогу виробника, що налічує більше 3000 найменувань.

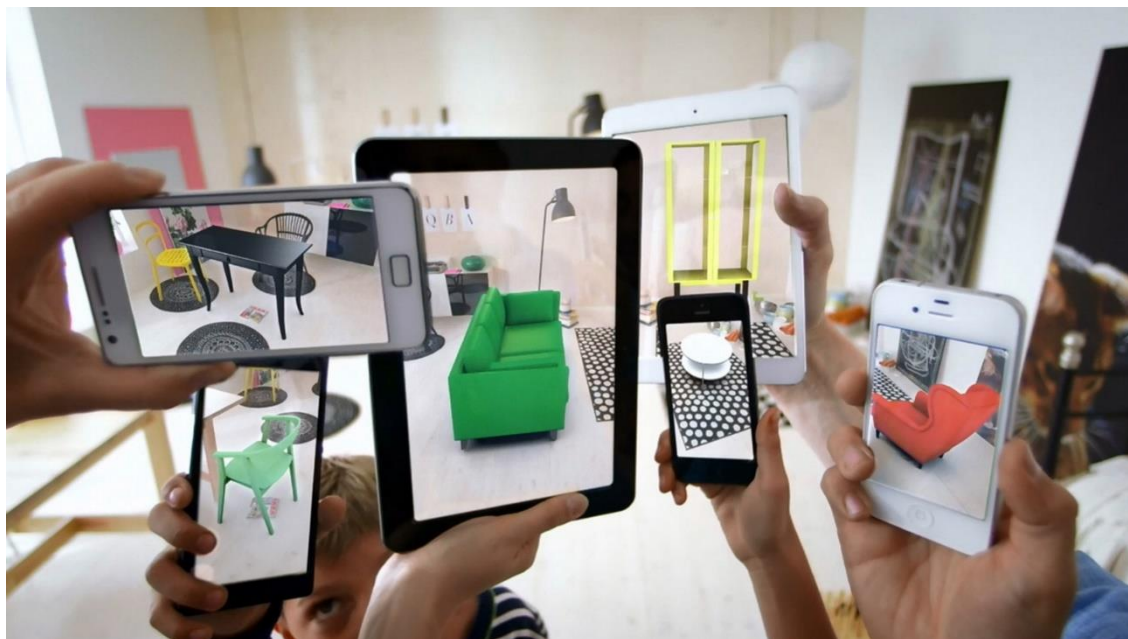


Рисунок 1.3 – Доповнена реальність у додатку IKEA Place

1.3 Доповнена реальність у інших галузях

1.3.1 Охорона здоров'я

Існує багато додатків доповненої реальності, які створені, щоб допомогти хірургам виконувати операції. AR зробила можливою операцію для менш досвідченого лікаря. Додаток Viipar в основному використовується хірургами-початківцями. Цей сервіс підтримує відео і може проектувати збільшені зображення рук на пацієнта, більш досвідчений хірург може направляти новачка за допомогою цих рук, перебуваючи навіть в іншому місці. Подібним прикладом є AssuVein, цей додаток використовує технологію AR для пошуку вен пацієнта.

1.3.2 Подорожі та туризм

AR-технології також використовуються у сфері туризму. Додаток Yelp Monocle, наприклад, використовує функцію AR для надання конкретної інформації про місцевість та місцезнаходження. За допомогою додатку користувач може направити свій телефон на певне місце, і функція AR надасть усю інформацію про це місце. Ця технологія доповненої реальності є корисною для людей, які подорожують новими локаціями без будь-якого путівника чи довідок. Використовуючи додатки AR на своєму смартфоні, вони можуть легко отримати доступ до місцевої інформації про будь-які місця.

1.3.3 Виробництво

Доповнена реальність охоплює також обробні галузі. Використовуючи додаток AR, працівники можуть легко організувати свої складальні лінії та склади. Сучасне виробництво передбачає точне складання сотень і тисяч деталей і для зручнішої організації цих інструкцій використовується технологія AR. Додаток Evolar використовується працівниками на складах для пошуку та зберігання пакунків, а також їх сортування та допомоги в обслуговуванні машин.

1.4 Аналіз існуючих рішень

Були вивчені існуючі системи доповненої реальності в різних сферах діяльності. Як приклади були представлені додатки, які працюють з різними типами пристроїв, мають різні технічні можливості. Кожен з них має свої переваги і недоліки, проаналізувавши які, можна визначити необхідні вимоги до створюваної системи.

Порівняльний аналіз представлено у Таблиці 1.1:

Таблиця 1.1 - Порівняльний аналіз існуючих рішень

	Мобільна версія (основна)	Веб-версія (допоміжна)	Мультиплатформенність	Оптимізація на сервері	Маркетплейс
Anatomy 4D	+	-	+	-	-
Word Lens	+	-	+	-	-
IKEA Place	+	-	+	+	+

Основні недоліки існуючих рішень:

- 1) відсутність єдиної системи для використання моделей;
- 2) відсутність маркетплейсу як найефективнішого рішення з реалізації продукту;
- 3) необхідність наявності додаткових ресурсів для використання додатків у інших сферах.

ВИСНОВОК ДО РОЗДІЛУ 1

У розділі 1 було розглянуто найбільш популярні серед користувачів системи доповненої реальності. Усі вони використовуються в різних сферах діяльності та за допомогою 3D-моделей у конструкціях доповненої реальності. Розглянуті сервіси використовують принцип користувацького вмісту та доступні у мобільних та веб-версіях. Після відбору та оцінки всіх факторів були зроблені висновки щодо основних загальних недоліків у сервісах та сформульовані вимоги до системи, які можуть усунути проблеми та підвищити ефективність. Основними недоліками вже реалізованих систем доповненої реальності є:

1) відсутність єдиної системи використання моделей - оскільки кожен сервіс використовує лише частину доступних моделей та виділяє лише одну сферу використання, ефективність пошуку та кількість знайдених рішень значно зменшуються;

2) відсутність охоплення якості обслуговування споживачів на місцевому рівні - маючи на увазі фокус на використанні послуг макромоніторингу, не забезпечується достатня вибірка даних для оцінки заявок різних типів;

3) слабо виражені, частково номінальні послуги сервісів на місцевому ринку - жоден з сервісів повністю не покриває потреб споживачів.

Отже, знайшовши основні недоліки кінцевих реалізацій, можна сформулювати вимоги до розвитку системи.

Вимоги до системи доповненої реальності:

1) можливість використання програми в різних сферах без використання додаткових ресурсів;

2) сучасний і зрозумілий інтерфейс для зручного використання системи;

3) створення оптимальної бази даних, яка відповідатиме потребам користувачів.

Таким чином, існує необхідність розробити сервіс для використання системи доповненої реальності, який відповідає всім вищезазначеним вимогам та усуває недоліки існуючих рішень. При створенні такого сервісу будуть використовуватися дві платформи розробки доповненої реальності, що значно розширює критерії та функції системи.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ДОДАТКУ

2.1. Опис предметної області

Доповнена реальність - це технологія, яка дозволяє доповнювати зображення реальних об'єктів різними об'єктами комп'ютерної графіки, а також співпрацювати із зображеннями, отриманими з різних джерел у обчислювальному середовищі [4].

За типом подання інформації системи доповненої реальності можна поділити на:

1. Візуальні, що засновані на зоровому сприйнятті людини. Завдання такої системи – створити образ для візуального сприйняття, яким користуватимуться люди. Оскільки зображення є більш інформативним і зрозумілим для людей, цей тип систем також є більш поширеним.
2. Аудіо - система, орієнтована на відтворення слуху. Такі системи використовуються в навігації - видають спеціальні сигнали, коли людина досягає певної локації. Можна використовувати стереоскопічний ефект, який дозволяє людині рухатися в правильному напрямку за допомогою звуку.
3. Змішана - поєднання попередніх двох типів, але аудіо-інформація в них виконує лише допоміжну функцію, коли основні функції має візуальна складова.

Системи доповненої реальності отримують дані із навколишнього середовища та будують віртуальні об'єкти на основі цих даних. Система має набір інструментів, що дозволяють збирати інформацію про навколишнє середовище. Ці інструменти можуть нести інформацію, відмінну від сигналів, тому за цим критерієм системи можна розділити на такі типи:

- AR на основі маркерів - розпізнавання зображень, для сканування яких потрібні візуальний об'єкт і камера. Використовуються QR-коди та спеціальні знаки. Пристрій доповненої реальності також

обчислює положення та орієнтацію маркера у просторі. Маркер запускає цифрові анімації, які користувач переглядає.

- AR без використання маркерів - доповнена реальність на основі геопозиціонування, яка використовує GPS, компас, гіроскоп та акселерометр для надання даних на основі місцезнаходження користувача. Ці дані визначають, які саме об'єкти доповненої реальності користувач може переглянути. Для смартфонів цей тип AR може створювати карти та надавати допомогу у навігації, інформацію про прилеглі локації.
- AR на основі проекції - проекція штучного світла на поверхні та взаємодія з ними. Це голограми, які виявляють взаємодію користувача з проекцією за її змінами.
- AR на основі суперімпозиції (накладання) - замінює фізичний об'єкт на доповнений за допомогою функцій розпізнавання об'єктів.

Системи доповненої реальності також відрізняються рівнем взаємодії з користувачем. У деяких системах користувач лише спостерігає відгук системи на зміни в середовищі. Інші системи вимагають активного втручання користувача - він може як керувати роботою системи для визначення результатів, так і вносити зміни до віртуальних об'єктів. На цій основі системи також поділяються на:

1. Автономні - системи, які не потребують втручання користувача. Основним завданням цих систем є надання інформації про об'єкти.
2. Інтерактивні – системи, засновані на взаємодії з користувачем - користувач отримує різні відповіді на дії.

В інтерактивних системах необхідно використовувати пристрій введення - можна використати сенсорний мобільний телефон, планшет або спеціальний пристрій.

У даній роботі система доповненої реальності інтегрована в структуру маркетплейсу для більшого охоплення сфер використання систем і підвищення працездатності сервісу.

Маркетплейс (marketplace) - це онлайн- платформа, яка допомагає продавцям та потенційним покупцям впровадити свої продукти до інтернет-ринку послуг та взаємодіяти між собою.

Основним фактором, за яким ідентифікується ринок, є наявність великої кількості компаній-постачальників товарів чи послуг, які виробляють та продають продукцію на єдиній платформі. З цієї причини ринок пропонує користувачам різноманітні пропозиції. Щоб ця система працювала належним чином, ринок повинен виконувати такі функції:

- познайомити споживачів з постачальниками;
- спростити для них процес комунікації, обміну товарами, послугами і платежами.

2.2 Визначення вимог і завдань

Основні функції сервісу:

- 1) Перегляд моделей доповненої реальності у мобільному додатку
- 2) Завантаження моделей в додаток
- 3) Створення опису моделей
- 4) Генерація QR-коду з посиланням на модель
- 5) Керування проектами в особистому кабінеті

Вимоги до системи

- 1) Наявність мультиплатформенного мобільного додатку для перегляду моделей
- 2) Можливість редагування вже завантажених моделей
- 3) Можливість завантаження моделей з серверу замість зберігання їх в додатку
- 4) Отримання швидкого доступу до розробників 3D-моделей (дизайнерів)

5) Категоризація проектів за типами.

2.3 Опис функціоналу системи

Система має надавати послуги користувачам з перегляду моделей в доповненій реальності, також розробникам 3D-моделей для створення і керування проектами.

Для користувачів надаються наступні можливості:

- 1) реєстрація нового користувача;
- 2) авторизація користувача за існуючою в базі профілів e-mail адресою і паролем;
- 3) перегляд і редагування моделі;
- 4) пошук моделі;
- 5) завантаження моделей;
- 6) генерація QR-коду;
- 7) створення проекту;
- 8) чат з власником моделі;
- 9) деавторизація.

2.4 Прецеденти

Після визначень основних вимог до системи і формування її функцій, необхідно побудувати сценарії використання системи та побудувати варіативну діаграму. На рисунку 2.1 зображена діаграма прецедентів сервісу, на якій показані всі основні взаємодії користувача з системою.

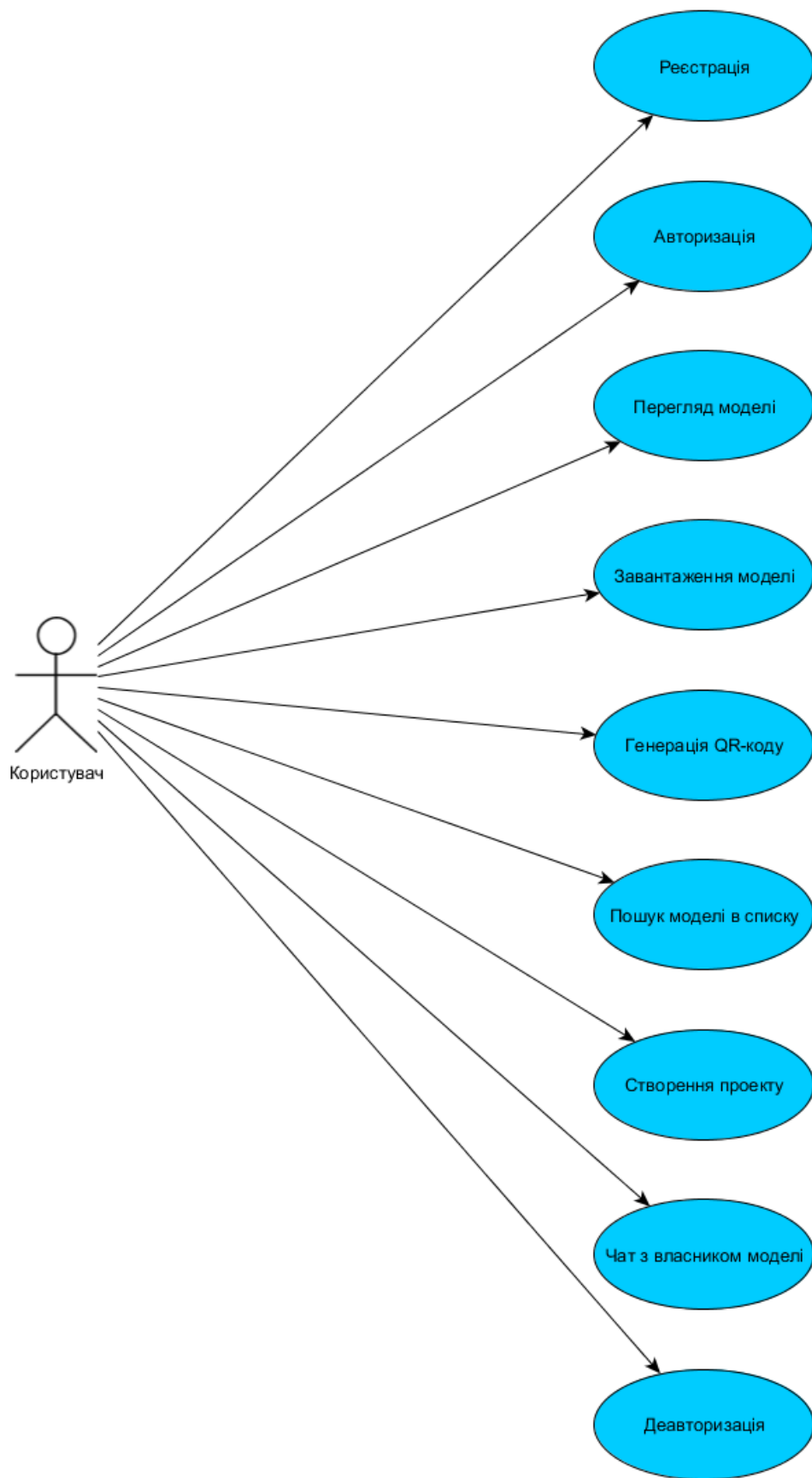


Рисунок 2.1 – Діаграма прецедентів системи

На рисунках 2.2- 2.6 зображено ієрархії прецедентів реєстрації та авторизації, пошуку моделі в списку, завантаження моделі і генерації QR-коду.

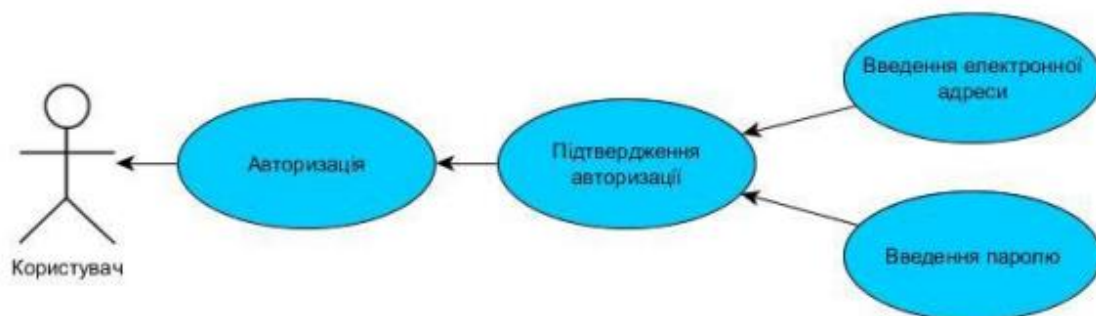


Рисунок 2.2 – Прецедент авторизації

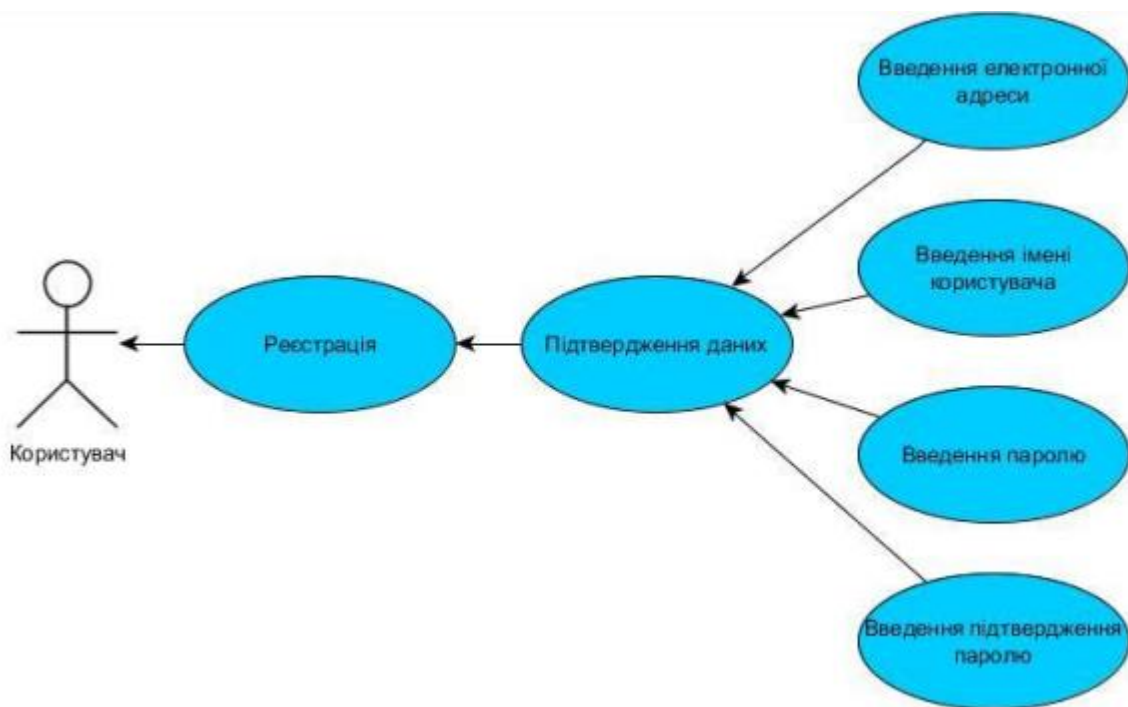


Рисунок 2.3 – Прецедент реєстрації

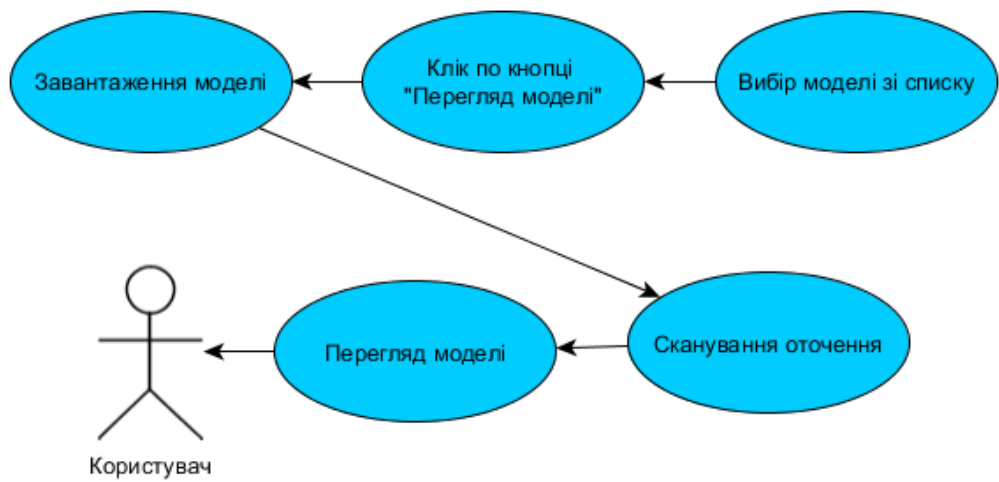


Рисунок 2.4 – Прецедент пошуку моделі в списку

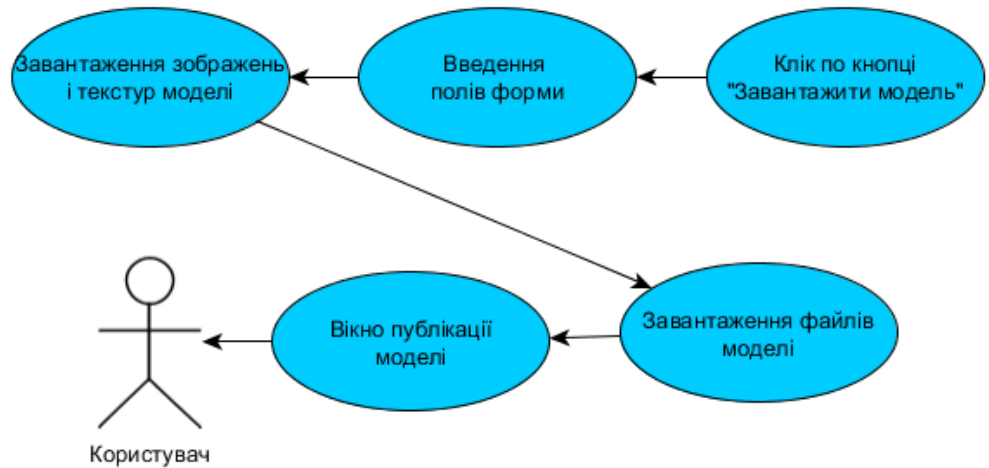


Рисунок 2.5 – Прецедент завантаження моделі

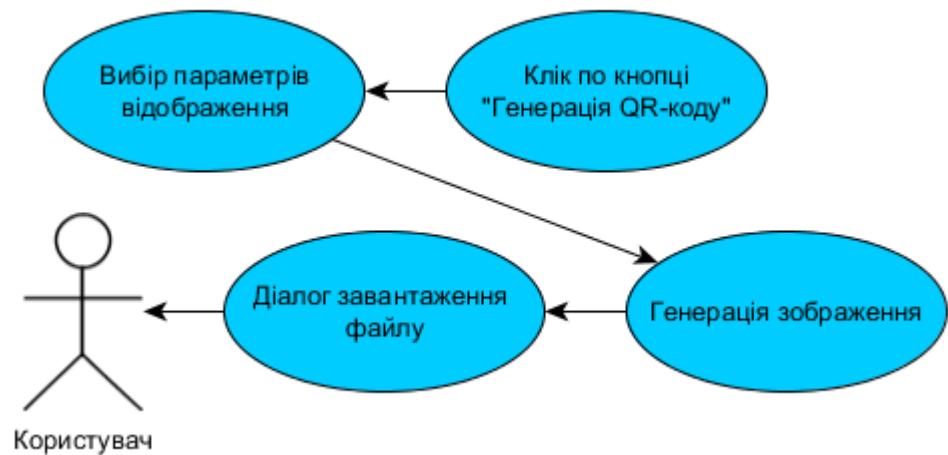


Рисунок 2.6 – Прецедент генерації QR-коду

2.4.1 Реєстрація

Таблиця 2.1 - Реєстрація

Дії користувача	Відгук системи
1) 1. Користувач вводить адресу електронної пошти, пароль, приймає умови користування та натискає кнопку «Авторизація»	2) Система перевіряє коректність введених даних
	3) Система створює новий профіль в базі даних та направляє користувача на сторінку авторизації

2.4.2 Авторизація

Таблиця 2.2 - Авторизація

Дії користувача	Відгук системи
1) 1. Користувач вводить адресу електронної пошти та пароль і натискає кнопку «Вхід».	2) Система відображає головну сторінку.

2.4.3 Перегляд моделі

Таблиця 2.3 - Перегляд моделі

Дії користувача	Відгук системи
1) Користувач обирає модель в списку	2) Система відображає користувачу екран з описом моделі.
3) Користувач натискає на кнопку «Перегляд моделі».	4) Система завантажує.

5) Користувач натискає кнопку «Сканування оточення»	6) Система починає сканування оточення
	7) Система відображає модель в доповненій реальності

2.4.4 Завантаження моделі

Таблиця 2.4 - Завантаження моделі

Дії користувача	Відгук системи
1) Користувач натискає на кнопку «Створити модель» на екрані проекту	2) Система відображає користувачу екран з формою додавання моделі.
3) Користувач заповнює дані форми	
4) Користувач завантажує текстури та файли моделі	5) Система завантажує файли на сервер та запускає процес конвертації моделі
	6) Система відображає діалогове вікно з опцією публікації моделі

2.4.5 Генерація QR-коду

Таблиця 2.5 - Генерація QR-коду

Дії користувача	Відгук системи
1) Користувач натискає на кнопку «Згенерувати QR код»	2) Система відображає користувачу екран з формою налаштування параметрів QR коду.
3) Користувач налаштовує QR код	
4) Користувач натискає на кнопку «Завантажити»	5) Система завантажує зображення QR-коду на комп'ютер користувача

2.5 Концептуальна діаграма класів структури даних

За допомогою прецедентів побудовано діаграму класів структури даних (рисунок. 2.7) .

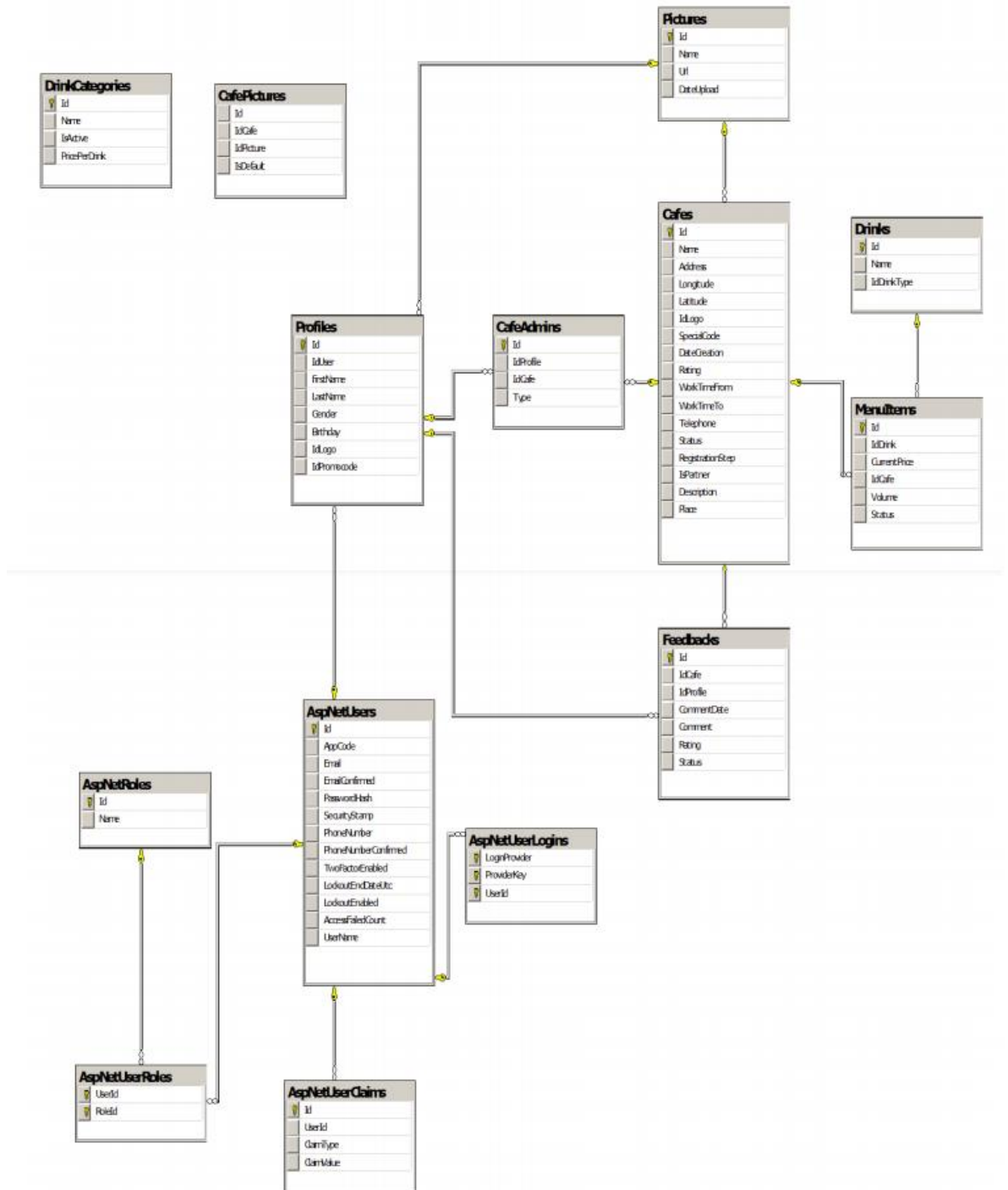


Рисунок 2.7 - Концептуальна діаграма класів структури даних

2.6 Проектування графічного інтерфейсу

2.6.1 Мобільний додаток

2.6.1.1 Головна сторінка

На головній сторінці мобільного додатку представлено:

- каталог у вигляді каруселі для пошуку 3D-моделей за призначенням; з функцією фільтрації ;
- розділ «популярне», у якому знаходяться моделі, що були завантажені найчастіше;
- кнопка «QR сканер» для сканування QR-коду моделей;
- кнопка «Повідомлення» для можливості контактування з розробниками моделей;
- кнопка «Обране» для збереження моделей, що сподобались.

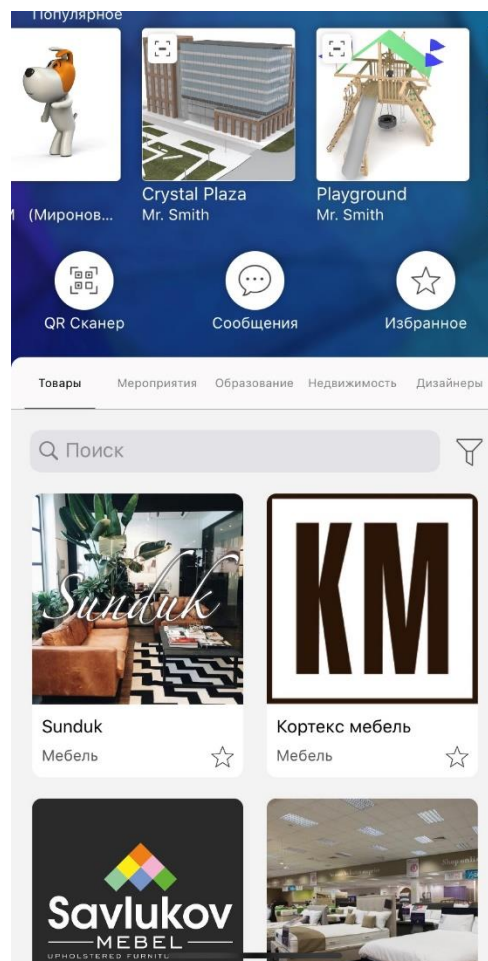


Рисунок 2.8 – Головна сторінка мобільного додатку

2.6.1.2 Сторінка 3D-моделі

На сторінці моделі представлено:

- попереднє зображення 3D-моделі;
- лейбл позиціонування 3D-моделі на поверхні;
- кнопка перегляду 3D-моделі (спочатку попередньо, потім - у режимі доповненої реальності);
- опис 3D-моделі;
- додаткові кнопки «Додому», «Поскаржитися», «Поділитися» та «Зберегти» відповідно.

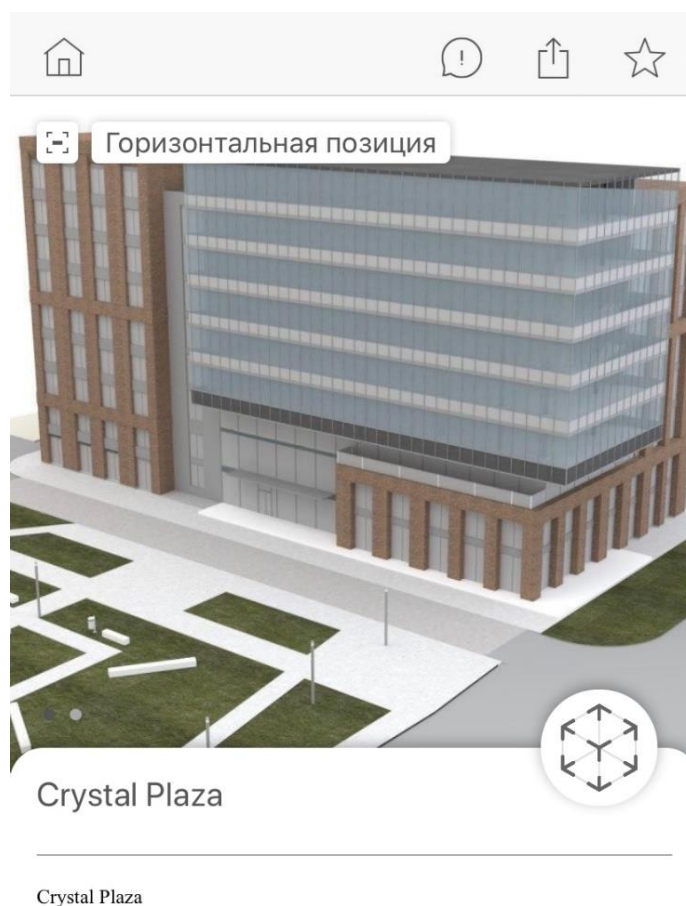


Рисунок 2.9 – Сторінка 3D-моделі у мобільному додатку

2.6.1.3 Чат

У чаті мобільного додатку користувач може відправити повідомлення користувачу веб-додатку (розробнику проекту).

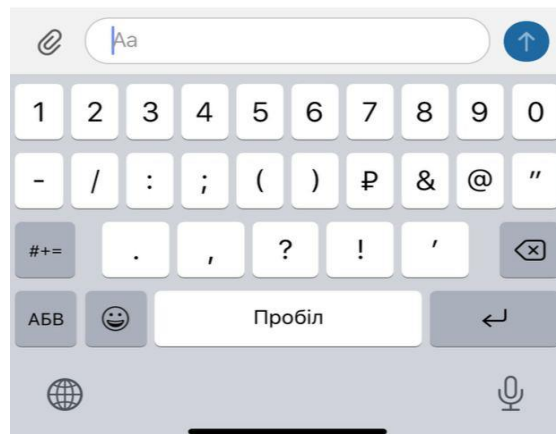


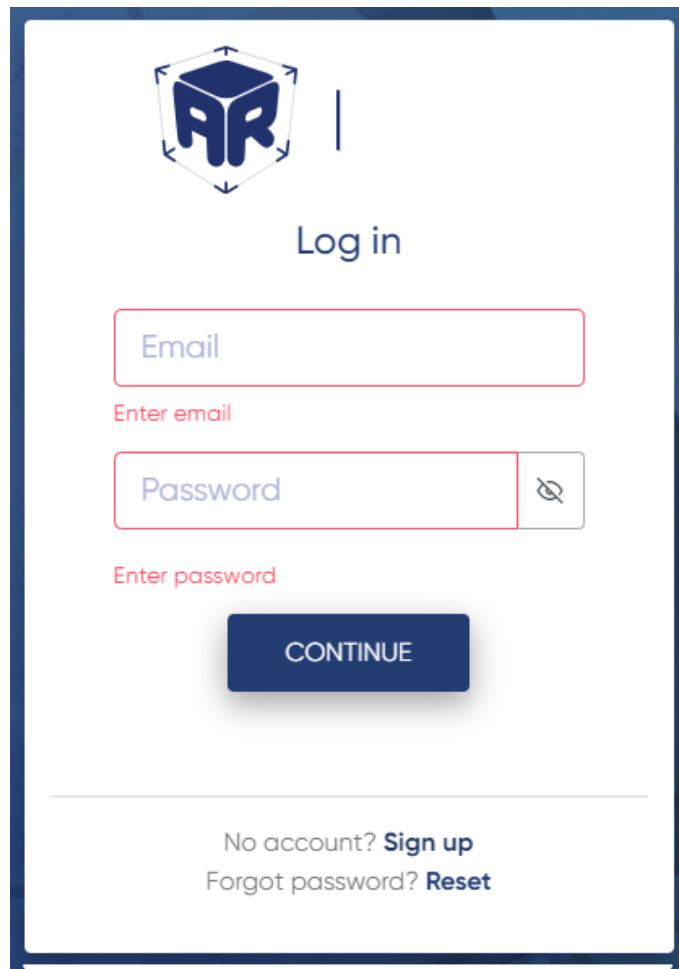
Рисунок 2.10 – Чат мобільного додатку

2.6.2 Веб-додаток

2.6.2.1 Вікно авторизації

Автоматично відображає поля для введення даних (адреси електронної пошти та пароля) для користувача. Система виконує перевірку на коректність введення і відображає повідомлення при помилці. При правильному введенні даних користувач повинен натиснути на кнопку «Продовжити» (Continue).

Також є кнопка «Забули пароль?» (Forgot password? Reset), яка перенаправляє користувача до вікна відновлення пароля і кнопка для переходу до вікна реєстрації для незареєстрованих користувачів (“Sign up”). При успішному проходженні валідації даних і натисканні на кнопку «Продовжити» система направляє користувача на головну сторінку веб-додатку і авторизує користувача в системі.



The screenshot shows a login window with a blue border. At the top center is a logo consisting of a blue cube with the letters 'AR' inside, followed by a vertical line. Below the logo is the text 'Log in'. There are two input fields: the first is labeled 'Email' with a red border and the placeholder text 'Enter email' below it; the second is labeled 'Password' with a red border, a placeholder text 'Enter password' below it, and a toggle icon (an eye with a slash) to its right. Below the password field is a blue button with the text 'CONTINUE'. At the bottom, there is a horizontal line, followed by the text 'No account? Sign up' and 'Forgot password? Reset'.

Рисунок 2.11 – Вікно авторизації у веб-додатку

2.6.2.2 Вікно реєстрації

Користувач вводить e-mail-адресу, пароль і підтверджує прийняття умов користування сервісом. Після успішної валідації даних і натискання кнопки «Продовжити » (Continue) користувач перенаправляється на головну сторінку.

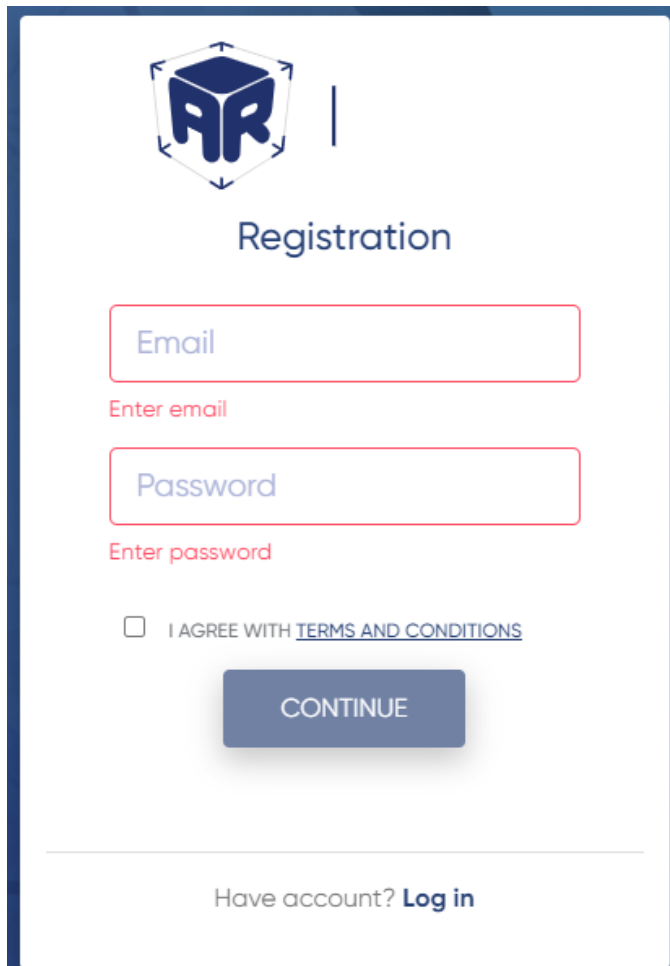
The image shows a registration form within a dark blue border. At the top center is a logo consisting of a blue cube with the letters 'AR' inside, followed by a vertical line. Below the logo is the title 'Registration' in a dark blue font. There are two input fields: the first is labeled 'Email' in light blue, with a red border and a red error message 'Enter email' below it; the second is labeled 'Password' in light blue, with a red border and a red error message 'Enter password' below it. Below these fields is a checkbox with the text 'I AGREE WITH [TERMS AND CONDITIONS](#)'. At the bottom center is a dark blue button with the word 'CONTINUE' in white. At the very bottom, separated by a horizontal line, is the text 'Have account? [Log in](#)'.

Рисунок 2.12 – Вікно реєстрації

2.6.2.3 Особистий кабінет

Меню особистого кабінету користувача складають таби:

- Статистика
- Мої проекти
- Повідомлення
- Підтримка

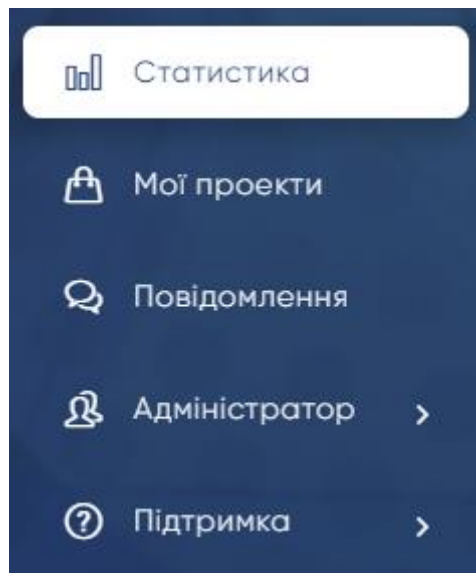


Рисунок 2.13 – Меню особистого кабінету

У табі «Статистика» користувач може переглянути:

- свій поточний тарифний план;
- стан сховища;
- графік та таблицю переглядів/ збережень/ часу перегляду в режимі AR та в прев'ю своїх 3D-моделей.

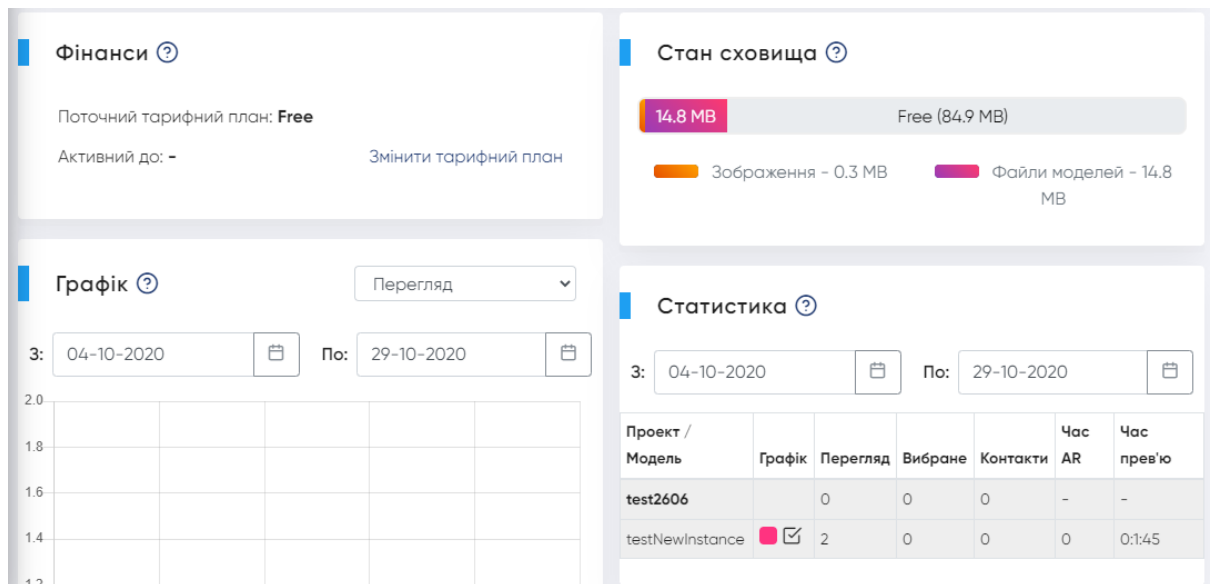


Рисунок 2.14 – Таб «Статистика» особистого кабінету

На сторінці проекту у табі «Мої проекти» користувач може:

- редагувати («Редагувати») проект;

- переглянути моделі в режимі прев'ю («Попередній перегляд з QR»);
- поділитися посиланням на проект («Поділитися»);
- приховати проект з табу «Мої проекти» («Приховати»);
- видалити проект («Видалити»);
- створити нову модель;
- редагувати опис моделей;
- видаляти моделі.

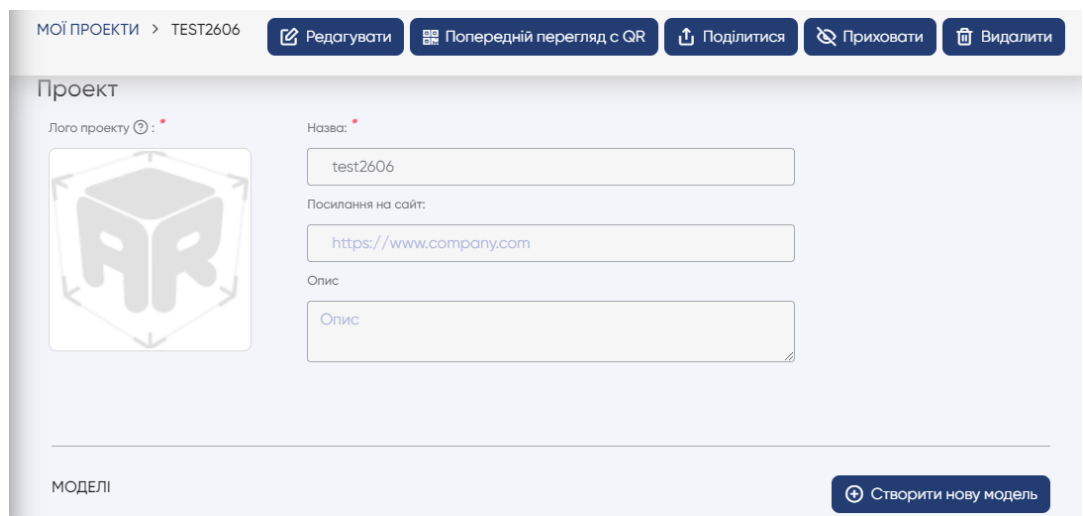


Рисунок 2.15 – Сторінка проекту

У табі «Повідомлення» користувач отримує повідомлення у чаті від користувачів мобільного додатку.

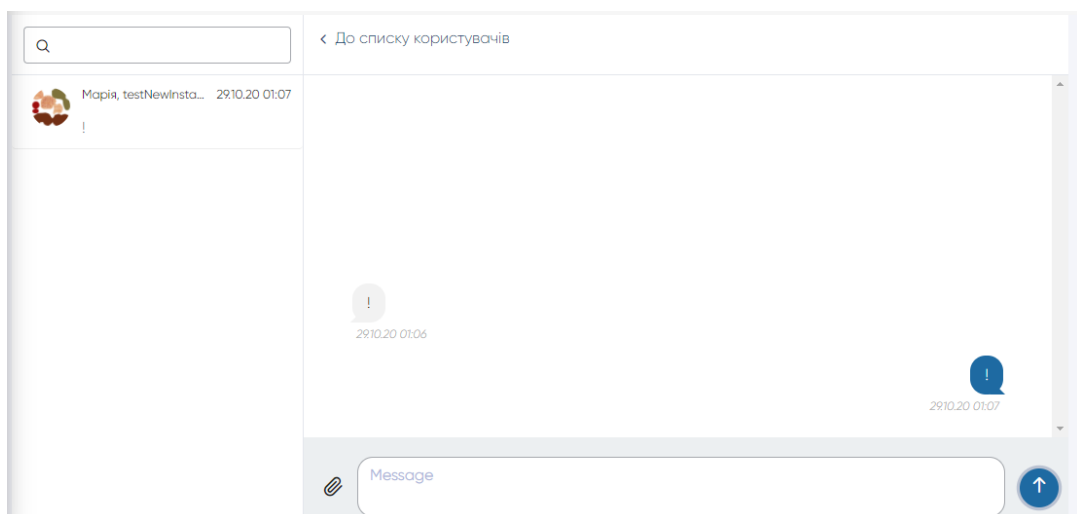


Рисунок 2.15 – Таб «Повідомлення»

ВИСНОВОК ДО РОЗДІЛУ 2

У розділі 2 була вивчена предметна область проекту, сформульовані основні функції системи і вимоги до неї.

Розроблено можливі варіанти сценаріїв використання системи, розглянуті відгуки системи при помилках, сценарії оповіщення користувача при некоректних вводах даних. За допомогою створених сценаріїв була побудована діаграма прецедентів множини функцій системи, а також окремі діаграми прецедентів для основних функцій системи:

- реєстрації користувача;
- авторизації користувача;
- перегляду і редагування моделі;
- пошуку моделі;
- завантаження моделей;
- генерації QR-коду;
- створення проекту;
- чату з власником моделі.

Був розроблений графічний інтерфейс сервісу, створена розмітка для основних сторінок і вікон:

- вікна реєстрації веб-додатку;
- вікна авторизації веб-додатку;
- головної сторінки веб-додатку (особистого кабінету);
- головної сторінки мобільного додатку;
- чату.

РОЗДІЛ 3.

РЕАЛІЗАЦІЯ СИСТЕМИ ДОПОВНЕНОЇ РЕАЛЬНОСТІ З ФУНКЦІЯМИ МАРКЕТПЛЕЙСУ

3.1 Архітектура системи

Як вже було описано вище, система має складатися з двох версій мобільних застосунків та веб-застосунку для завантаження моделей. Для організації взаємодії застосунків та збереження даних доцільно розробити WEB API та використовувати реляційну БД, адже такий тип баз даних найбільше підходить для даного типу системи, яка має строгу структуру даних та має швидко опрацьовувати великі масиви даних. Для збереження моделей доцільно використовувати файлову систему сервера.

Кінцева архітектура системи зображена на рисунку:

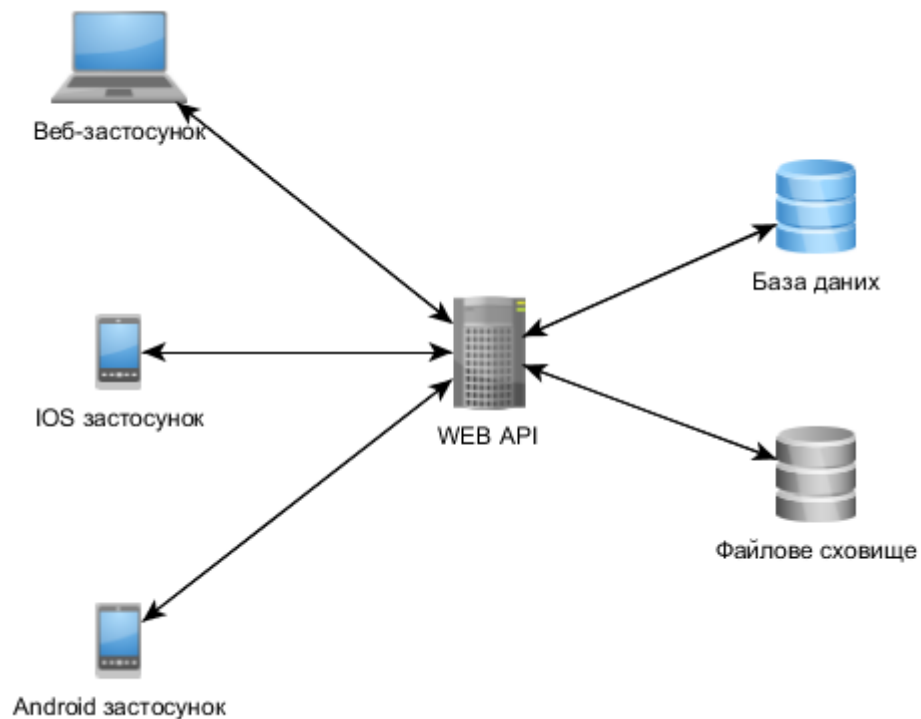


Рисунок 3.1 – Архітектура системи

3.2. Вибір технологій та їх обґрунтування

3.2.1 Технології для реалізації серверної частини

Для реалізації серверної частини необхідно визначитись з технологіями для розробки API та типом баз даних.

Для реалізації API популярними мовами програмування є C# та Python. Необхідно розглянути їх більш детально.

C# - це потужна мова, яка точно відповідає традиційним конструкціям C & C++, але вона є більш сучасною та легшою для вивчення. Розроблена Microsoft, ця об'єктно-орієнтована мова програмування також має багато спільного з Java [5]. Код C# може бути скомпільований на різних платформах і має безліч потужних функцій, таких як -

- Інтеграція з .NET Framework
- Структурована мова високого рівня
- Сучасний синтаксис
- Багата стандартна бібліотека
- Автоматичне вивільнення пам'яті

Основна структура програми на C# подібна до структури на C++ та Java. Оголошення простору імен, визначення класу (змінні та методи), основний метод - і все. Ось проста програма, яка друкує ім'я користувача.

Подібно до C#, Python є мовою програмування загального призначення. Він дотримується C & Java у більшості своїх функціональних можливостей. Це портативна та проста у вивченні мова з високими можливостями програмування [6]. Це мова, що інтерпретується (тобто перевірка типу здійснюється під час виконання).

Деякі особливості Python:

- Підтримує об'єктно-орієнтоване програмування, а також функціональне та структуроване програмування
- Легко кодувати, читати, обслуговувати
- Велика стандартна бібліотека, портативна та сумісна з різними платформами, таких як Windows, Mac або Unix.
- Підходить для мережесхемних додатків, що використовують кілька протоколів.

Виходячи з перерахованих вище переваг, було прийнято рішення використовувати мову C# для розробки API.

MySQL - одна з найпопулярніших баз даних для веб-додатків. MySQL - це безкоштовний програмний пакет, але постійно виходять нові версії, що розширюють функціональність та покращують безпеку. В наявності є платна версія для використання в комерційних проектах. У безкоштовній версії найбільший акцент робиться на швидкості та надійності, а не на повноті функціональності, яка може бути як плюсом, так і мінусом, залежно від сфери впровадження [7].

MySQL розробляється та підтримується корпорацією Oracle, яка придбала права на торгову марку спільно із придбаною Sun Microsystems, яка раніше придбала шведську компанію MySQL AB. Продукт поширюється як за загальною публічною ліцензією GNU, так і за власною комерційною ліцензією. Крім того, розробники створюють функціонал, упорядкований ліцензованими користувачами. Саме завдяки цьому наказу механізм реплікації з'явився майже в перших версіях.

Ця СУБД дозволяє вибирати різні механізми зберігання, які дозволяють змінювати функціональні можливості інструменту та обробляти дані, що зберігаються в різних типах таблиць. Завдяки відкритій архітектурі та ліцензіям GPL у MySQL постійно з'являються нові типи таблиць. MySQL також має простий у використанні інтерфейс та пакетні команди, які дозволяють легко управляти величезними обсягами даних. Система

неймовірно надійна і не намагається підпорядкувати собі всі наявні апаратні ресурси.

Переваги:

- безкоштовне розповсюдження;
- відмінно задокументована;
- пропонує безліч функцій, навіть у безкоштовній версії;
- пакет MySQL входить до стандартних сховищ найпоширеніших дистрибутивів Linux, що полегшує його встановлення;
- підтримує набір користувальницьких інтерфейсів;
- може працювати з іншими базами даних, включаючи DB2 та Oracle.

Недоліки:

- Для виконання простих завдань MySQL займе багато часу та зусиль, хоча інші системи роблять це автоматично, наприклад: створюють додаткові резервні копії.
- Немає власної підтримки XML або OLAP.
- Для безкоштовної версії доступна лише платна підтримка.

Microsoft SQL Server - це система управління базами даних, ядро якої працює на хмарних серверах, а також на локальних серверах, і дає змогу поєднувати поєднувати типи серверів, що використовуються одночасно. Незабаром після випуску Microsoft SQL Server 2016, Microsoft адаптував продукт під операційну систему Linux, спочатку він працював лише на платформі Windows.

Переваги

- продукт дуже простий у використанні;
- поточна версія швидка та стабільна;

- дозволяє регулювати та контролювати рівні продуктивності, які допомагають зменшити використання ресурсів;
- добре взаємодіє з іншими продуктами Microsoft.

Недоліки:

- Ціна для юридичних осіб неприйнятна для більшості організацій.
- Навіть при ретельному налаштуванні продуктивності, SQL Server може споживати всі доступні ресурси.
- Повідомлялося про проблеми за допомогою служби інтеграції для імпорту файлів.

PostgreSQL - одна з багатьох безкоштовних та популярних систем управління базами даних, які часто використовуються для управління веб-сайтами баз даних. Це одна з перших розроблених систем управління базами даних, тому зараз вона має велику підтримку і дозволяє користувачам керувати структурованими та неструктурованими даними [8]. Його можна використовувати на більшості основних платформ, включаючи Linux. Ідеально справляється із завданнями імпорту інформації з інших типів баз даних за допомогою власних інструментів.

Механізм бази даних може розміщуватися в різних середовищах, включаючи віртуальне, фізичне та хмарне. Остання версія, PostgreSQL 9.5, пропонує велике управління даними та збільшення кількості одночасних користувачів. Безпека була покращена завдяки підтримці DBMS_SESSION.

Переваги:

- PostgreSQL масштабований і здатний обробляти терабайти даних.
- Підтримує формат JSON.
- Є багато заздалегідь реалізованих функцій.

Недоліки:

- Документація розмита, можливо, доведеться шукати в Інтернеті відповіді на деякі питання.

- Конфігурація може заплутати недосвідченого користувача.

- Продуктивність може бути повільною під час пакетних операцій або запитів на читання.

Останній варіант ідеально підходить для організацій з обмеженим бюджетом, але кваліфікованих фахівців. Тому було вирішено використовувати в якості СУБД для проекту PostgreSQL. Тим більше, що .Net Core має ORM-провайдера для цієї СУБД. Таким чином, взаємодія з базою даних значно спроститься.

3.2.2 Технології для реалізації мобільних додатків

ARKit, ARCore та Vuforia - одні з найпопулярніших наборів інструментів, які необхідні для створення додатка з доповненою реальністю. Незважаючи на те, що всі вони можуть запропонувати хороші можливості, є незначні моменти, коли розробники повинні знати і враховувати, вибираючи відповідні інструменти, фреймворк та SDK для розробки додатків AR

ARKit від Apple - це набір інструментів (фреймворк та SDK), які допомагають розробникам створювати AR-програми для платформи iOS для включення моделей iPhone 6 і вище та моделей iPad Pro [9].

Основними перевагами ARKit є виявлення площини, карта оточення AR, оцінка світла, якорі, відстеження обличчя, захоплення руху.

ARKit підтримує виявлення зображень та відстеження зображень та допомагає інтегрувати віртуальні об'єкти в AR. Головною перевагою ARKit для розробників перед іншими популярними інструментами розробки AR є екосистема Apple, яка полегшує розробку. Крім того, він пропонує часті оновлення та нові функції для більш цікавих можливостей розробки додатків.

Але, як уже зазначалося, суттєвим обмеженням ARKit для розробки додатків є той факт, що його використання обмежене платформою iOS. Крім того, він не справляється із захопленням руху при дуже швидких рухах.

ARKit - це інструментарій Google (фреймворк та SDK) для створення додатків з доповненою реальністю

ARCore пропонує функціонал виявлення площини, оцінку освітленості, якорі, відстеження зображень, відстеження обличчя ті інші. Ці можливості та результати подібні до ARKit [10].

Використовуючи можливості відстеження руху ARCore, розробники можуть відстежувати положення телефону щодо навколишнього середовища. Іншими важливими можливостями ARCore є розуміння навколишнього середовища, включаючи визначення розміру та розташування поверхонь та оцінку освітленості, включаючи фактичні умови освітлення.

Завдяки можливостям відстеження позицій у реальному часі та інтеграції віртуальних та реальних об'єктів, ARCore є чудовим інструментом для AR у додатках електронної комерції [11].

Vuforia підтримує Android, iOS, UWP та Unity Editor. Цей набір інструментів розвитку PTC AR - ще один хороший варіант для розробників додатків AR для відстеження заздалегідь визначених 3D-зображень, моделей, об'єктів або сканувань.

Основними особливостями Vuforia є відстеження об'єктів та зображень для розробки AR. Платформа Vuforia пропонує виявлення об'єктів із цілей, передбачених для розпізнавання, і таким чином дозволяє розробникам завантажувати моделі, зображення, сканування об'єктів та інші типи допоміжних цілей для виявлення [12].

Серед недоліків Vuforia є висока ціна за використання (99\$/місяць), недостатня документованість системи, та наявність багатьох багів та проблем, що призводять до збоїв в роботі застосунків.

ARKit, ARCore та Vuforia - популярні набори інструментів для розробників AR. Використання Vuforia в даній системі було відкинуто через її недоліки. Оскільки ARCore працює найкраще для Android, а ARKit надає нативні можливості в IOS були вирішено, що для обох систем будуть використовуватися різні фреймворки, для того, щоб мати більший контроль над системою, та мати змогу якнайкраще реалізувати можливості доповненої реальності для кожної з систем.

3.2.3. Технології для реалізації веб-застосунку

React, розроблений Facebook, був випущений в 2013 році. Facebook активно використовує React у своїх продуктах (Facebook, Instagram та WhatsApp). Поточна стабільна версія - 16.X, випущена в листопаді 2018 року.

React здається хорошим вибором для когось, хто новачок у фронтендних фреймворках JavaScript, стартапах та для розробників, яким подобається гнучкість. Можливість плавної інтеграції з іншими фреймворками дає велику перевагу над іншими рішеннями [13].

Angular, розроблений Google, був випущений в 2010 році, що робить його найстарішим із групи. Це фреймворк JavaScript, заснований на TypeScript. Суттєва зміна відбулася в 2016 році з випуском Angular 2 (і вилученням "JS" з початкової назви - AngularJS). Angular 2+ відомий як просто Angular. Хоча AngularJS (версія 1) все ще отримує оновлення, ми зосередимо обговорення на Angular. Останньою стабільною версією є Angular 10, яка вийшла в червні 2020 року.

Angular - хороший вибір для підприємств з великими командами та розробниками, які вже використовують TypeScript [14]. Саме його було обрано для розробки веб-застосунку.

3.3. Реалізація WEB API

В наступній таблиці представлені методи WEB API, що використовуються для взаємодій додатків з сервером.

Таблиця 3.1 – Список методів WEB API

Назва методу	Призначення методу
/api/v1/AppConfiguration	Отримання даних конфігурації веб-додатку
/api/v1/CategoriesGet	Отримання списку категорій та підкатегорій для продуктів
/api/v1/SearchCompanies	Метод для пошуку компаній/проектів
/api/v1/SearchProducts	Метод для пошуку продуктів
/api/v1/CompaniesGet	Метод для отримання списку компаній/проектів
/api/v1/CompanyGet	Метод для отримання даних по одному проекту
/api/v1/ProductsGet	Метод для отримання списку продуктів
/api/v1/ProductGet	Метод для отримання даних по одному продукту
/api/v1/RecentGet	Метод для отримання нещодавно доданих в систему моделей, для відображення в верхній панелі
/api/v1/SendProductStatistics	Метод для відправки статистики по продукту
/api/v1/SendCompanyStatistics	Метод для відправки статистики по компанії/проекту
/api/v1/SendCompanyReport	Метод для відправлення скарги на компанію/проекту

/api/v1/SendProductReport	Метод для відправки скарги на продукт
/Storage/LoadModel	Метод для завантаження файлів моделу

POST /api/v1/AppConfiguration – метод для отримання конфігурації додатку при відкритті додатку. Параметри відсутні. У відповідь відправляється об'єкт з полями:

"companyCount" – кількість компаній в системі, якщо відмінне від того, що збережене в додатку, потрібно синхронізувати дані

"categoriesCount": – кількість категорій в системі, якщо відмінне від того, що збережене в додатку, потрібно синхронізувати дані

"requiredArcoreVersion": - версія Arcore, додаток на Android використовує це поле, для того, щоб перевіряти версію, і, якщо вона відмінна від версії в додатку, виводити повідомлення про необхідність оновлення додатку.

POST /api/v1/CompaniesGet – метод для отримання списку компаній/проектів. Параметри name, companyTypeIDs, categoryIDs – використовуються для фільтрації списку компаній.

У відповідь надсилається масив об'єктів з наступною структурою:

"id" – ідентифікатор компанії/проекту

"companyTypeID" – тип компанії/проекту

"name" – назва

"siteLink" – посилання на сайт

"description" – опис компанії/проекту

"mainCategories" – список з категоріями, використовується для пошуку та фільтрації

/api/v1/ProductGet – метод для отримання даних по одному конкретному продукту. Параметри: productID – ідентифікатор продукту. У відповідь надсилається об'єкт з наступною структурою:

"id" – ідентифікатор продукту

"name" – назва продукту

"arScanType" – тип AR – розміщення на маркері, розміщення на вертикальній поверхні, розміщення на горизонтальній поверхні

"modelUnit" – одиниці виміру для обрахування масштабу модулі

"productCode" – код продукту (штрих-код), опціональне поле

"images" – масив з посиланнями на зображення продукту

"markerImages" – масив з посиланнями на зображення маркерів, для продукту з типом відображення на маркері. Ці зображення використовуються для розпізнавання маркерів при скануванні оточення.

"modelFileName" – назва директорії, що містить файли моделі

"description" – опис моделі

"shinyMaterials" – значення в діапазоні 0-100, що відповідає за ступінь віддзеркалення оточення від поверхні моделі (0 задає матову поверхню).

"contactID" – ідентифікатор контакту, що зв'язаний з моделлю

"companyID" – ідентифікатор компанії

"companyName" – назва компанії

"libCategoryID" – категорія продукту

"subCategoryID" – підкатегорія продукту

"subCategoryName" – назва підкатегорії

"markerWidth" – для моделей з типом відображення на маркері, задає ширину маркера в см

"status" – статус моделі (опублікована, прихована, видалена)

3.4. Реалізація мобільних додатків

Мобільний додаток працює без авторизації, таким чином спрощується взаємодія користувача з застосунком. При вході в додаток, користувач одразу потрапляє на головну сторінку, яка складається з декількох секцій. Перша секція – список з нещодавно доданих моделей. Друга містить кнопки для швидкого доступу до функцій сканування маркерів, перегляду повідомлень та перегляду моделей чи проектів, що були додані в «Обране». Внизу знаходиться секція зі списком проектів та елементами пошуку та фільтрації.

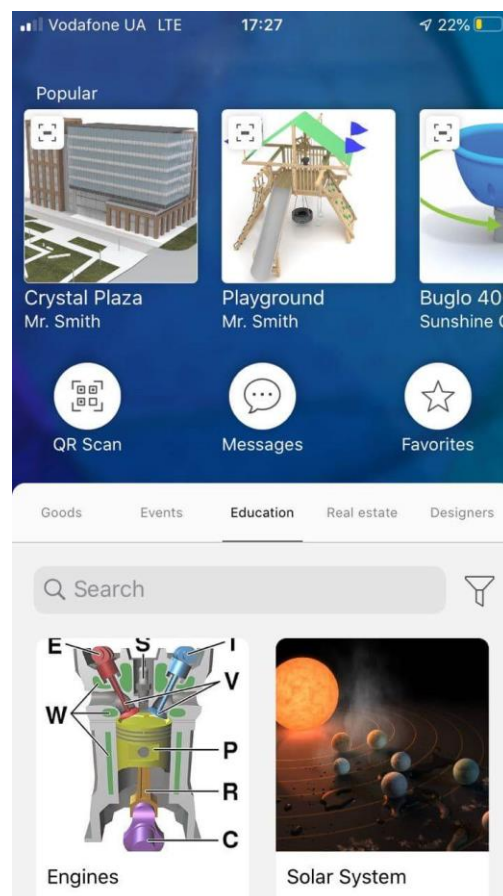


Рисунок 3.2 – Головна сторінка мобільного додатку

З секції зі списком проектів користувач має змогу перейти до екрану проекту. На цій сторінці міститься інформація про проект, список моделей цього проекту та кнопок для декількох функцій. Цими функціями є:

- відправлення скарги на проект, якщо, на думку користувача, проект не відповідає нормам чи порушує авторські права
- кнопка для відправки посилання на проект
- кнопка для додавання проекту в «Обране»

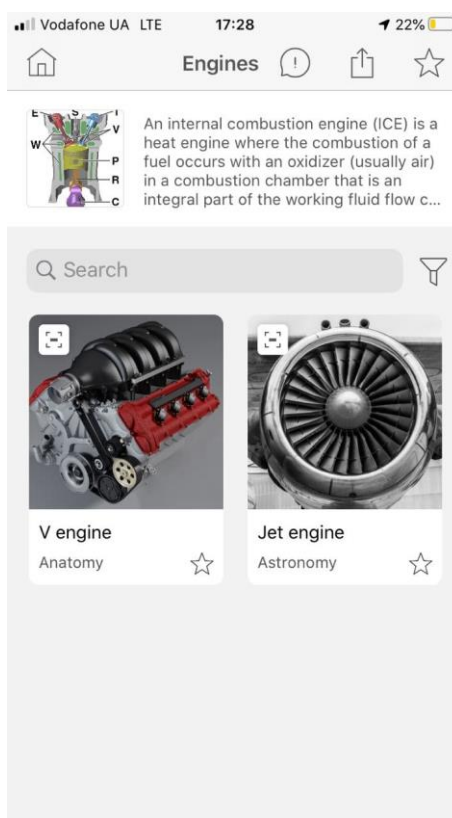


Рисунок 3.3 – Сторінка проекту в мобільному додатку

Зі сторінки проекту можна перейти на екран перегляду конкретної моделі. За дизайном, ця сторінка схожа на сторінку проекту, але містить додаткові функції, такі як кнопка для перегляду моделі в режимі доповненої реальності та кнопка для чату з власником моделі. Після кліку на перегляд моделі в доповненій реальності, починається завантаження моделі з сервера. Користувачу в цей час відображається екран, що демонструє процес завантаження моделі (у відсотках). Ця функція покращує UX застосунку,

адже деякі моделі є громіздкими, та мають видиму затримку під час завантаження. За допомогою цього екрану користувач має змогу оцінити час завантаження моделі.



Рисунок 3.4 – Головна моделі в мобільному додатку

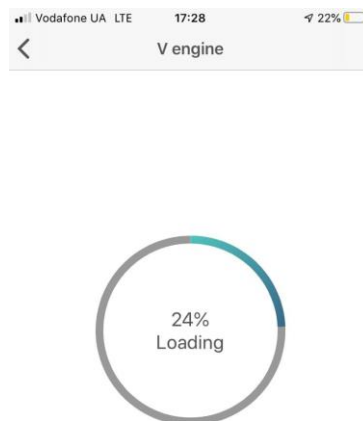


Рисунок 3.5 – Екран завантаження моделі

Після завантаження моделі відкривається сторінка з попереднім переглядом моделі та кнопкою, що запускає сканування оточення. Під час сканування оточення користувач захопити камерою простір, на якому буде розміщена модель, або навести камеру на маркер, до якого модель буде прикріплена. Цей процес супроводжується підказками. Після того, як модель зафіксована, користувач має змогу взаємодіяти з нею в режимі доповненої реальності та знімати фото/відео з захопленням екрану, при цьому модель буде в кадрі.



Рисунок 3.6 – Екран попереднього перегляду моделі

Ще одним важливим екраном в мобільному застосунку є екран чатів, в якому користувач може надсилати текстові повідомлення та зображення власнику моделі. Таким чином відбувається комунікація між клієнтами мобільних та веб-застосунку.



Рисунок 3.7 – Сторінка чату в мобільному додатку

3.5. Реалізація веб-клієнту

Користувача в системі зустрічає форма логіну, яку він повинен заповнити для переходу в застосунок. Також, якщо користувач ще не має аккаунту, на цій формі є кнопка для переходу на форму реєстрації.

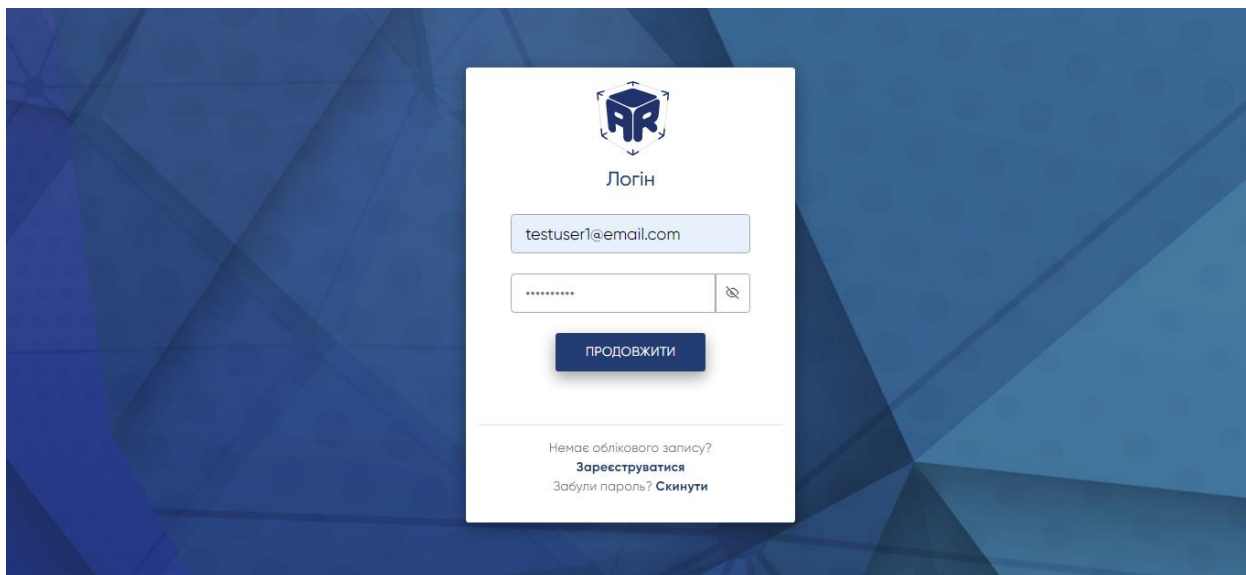


Рисунок 3.8 – Сторінка авторизації в веб-застосунку

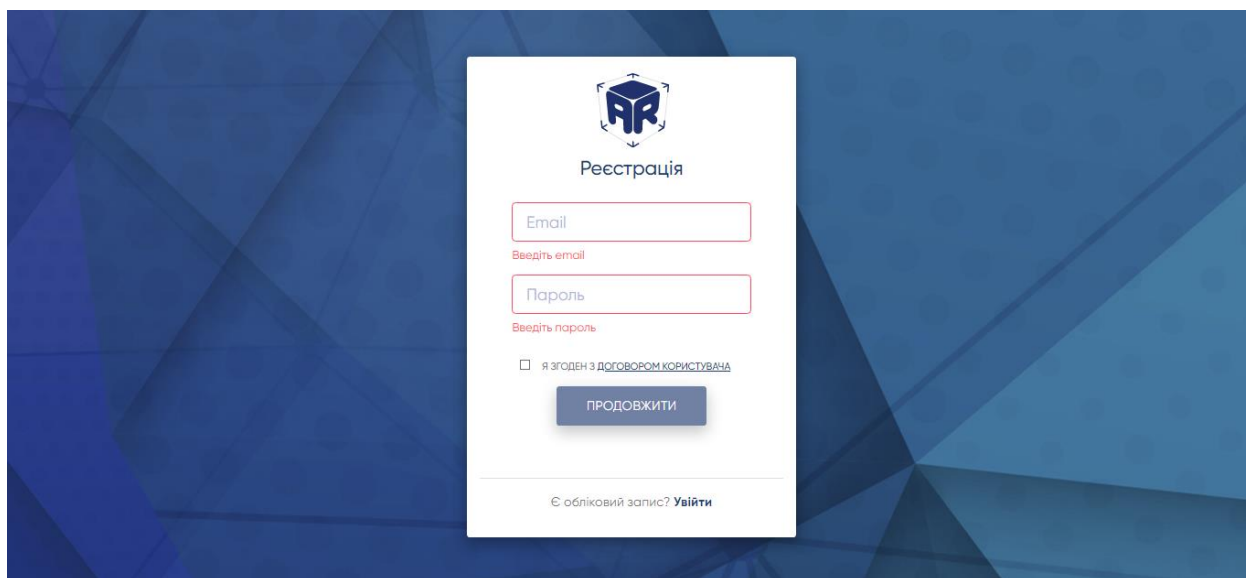


Рисунок 3.9 – Сторінка реєстрації в веб-застосунку

Після авторизації, користувач потрапляє на головну сторінку, на якій виводяться дані про стан підписки, сховища, а також статистичні дані про перегляд проектів/моделей у табличному вигляді та на графіці.

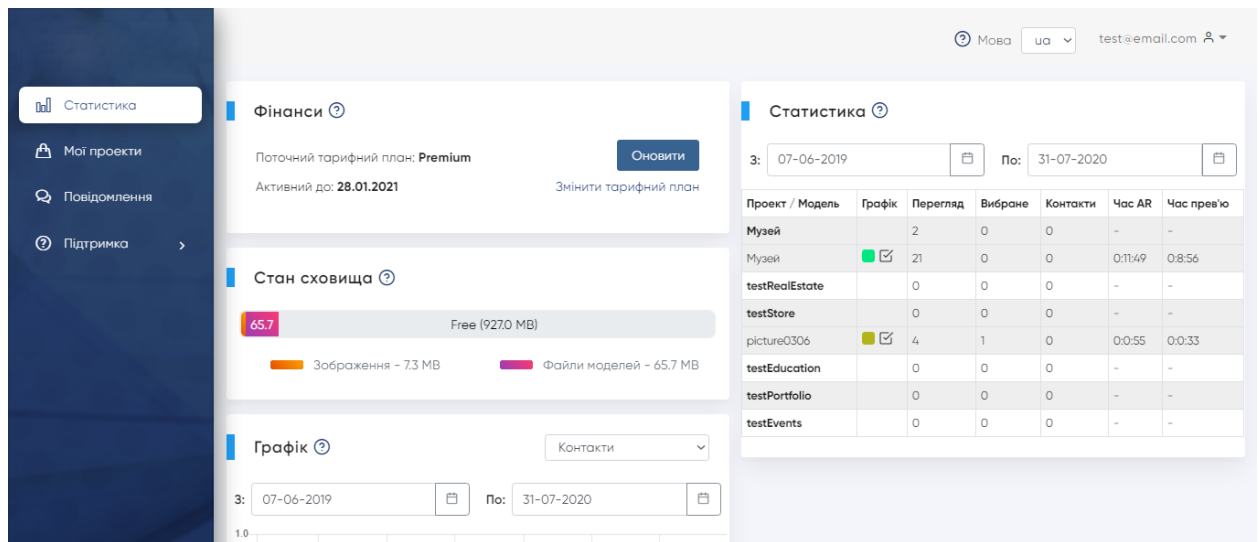


Рисунок 3.10 – Головна сторінка веб-застосунку

Наступним елементом меню є список проектів, на який виводяться всі проекти, доступні користувачу, з вказанням розміру, який проект займає в файловому сховищі. При цьому проекти, що не опубліковані, виводяться з відповідною іконкою, що вказує його стан. Зі списку проектів користувач може перейти на сторінку конкретного проекту, де має можливість продивитись моделі, які належать цьому проекту, відредагувати дані, приховати, опублікувати чи видалити проект, а також згенерувати QR-код, що буде посилатися на цей проект і відкривати його в мобільному додатку.

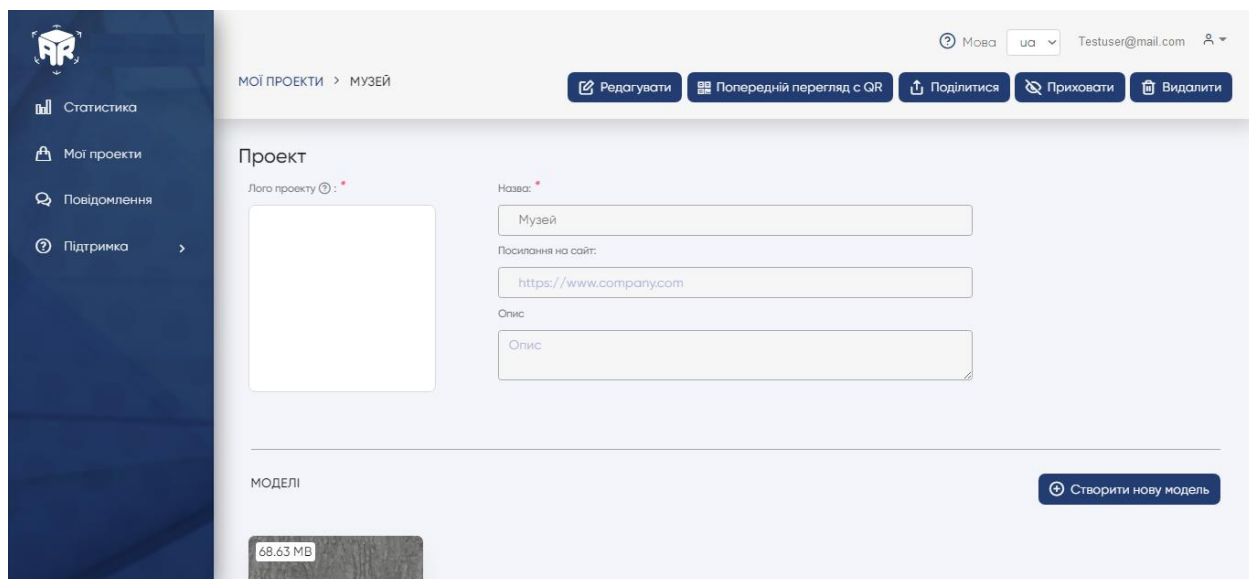


Рисунок 3.11 – Сторінка проекту в веб-застосунку

Для створення нової моделі, користувач може натиснути на відповідну кнопку на сторінці проекту, та перейти на форму створення моделі. Ця форма поділена на логічні блоки, деякі з яких залежать від того, який тип відображення в доповненій реальності буде використовуватися для конкретної моделі. Для кращого UX є спеціальну блоки, які демонструють вплив параметрів на кінцеве відображення моделей. Також, для спрощення роботи, є кнопка, за допомогою якої можна завантажити приклад файлів моделей, для того, щоб перевірити відповідність нормам валідації. Після створення моделі, користувач переходить на сторінку перегляду моделі, яка є досить подібною до сторінки проекту. Тут є ті ж самі функції, але вони діють вже відносно моделі.

Статистика

Мої проекти

Повідомлення

Підтримка

Мова

ua

98sancho@ukr.net

МУЗЕЙ > СТВОРИТИ НОВУ МОДЕЛЬ

Зберегти

Відміна

Основне

Назва *

Назва продукту

Код продукту *

Код продукту

Опис *

Введіть текст...

Контакти

Посилання на продукт:

Instagram:

Телефон:

+

Telegram:

Email:

Viber:

+

Категорії моделі

Загальна категорія: *

Виберіть загальну категорію

Користувачка категорія: *

Виберіть загальну категорію

Налаштування моделі

Тип AR: *

Одиниці виміру: *

Виберіть одиниці виміру моделі

Рівень блиску: *

100

Зображення

Файли моделі

Завантажити приклад

Формат .obj

Формат .mtl

Текстури (усі файли)

Рисунок 3.12 – Сторінка створення моделі в веб-застосунку

Для комунікації з користувачами мобільних додатків є окремий розділ, який має інтерфейс зі списком чатів, та є подібним до багатьох реалізацій чатів в соціальних месенджерів. Користувач має змогу відправляти як текстові повідомлення, так і зображення.

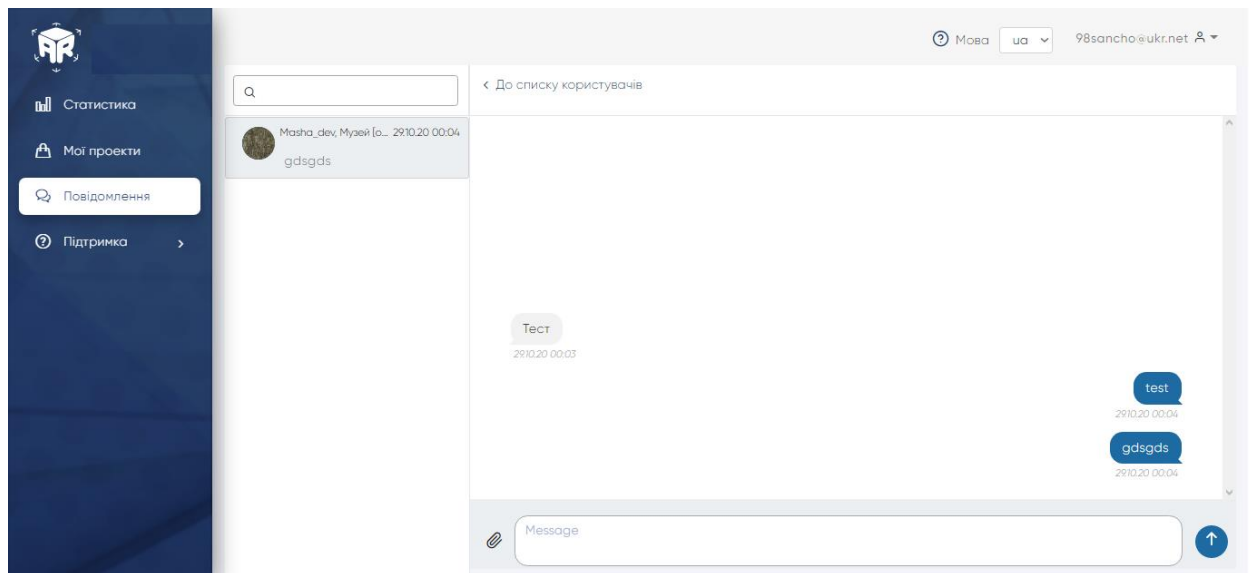


Рисунок 3.13 – Сторінка з чатами в веб-застосунку

ВИСНОВОК ДО РОЗДІЛУ 3

В цьому розділі було описано процес розробки програмної частини системи доповненої реальності з функцією маркетплейсу. В ході розробки системи було вирішено наступні завдання:

1. Було обрано технології для розробки частин системи
2. Реалізовано методи API, які використовуються для взаємодії клієнтів мобільних та веб-додатків з сервером.
3. Реалізовано мобільні застосунки на IOS та Android для клієнтів системи. Налаштовано завантаження моделей з серверу та їх відображення в режимі доповненої реальності.
4. Реалізовано веб-застосунок для завантаження та менеджменту моделей.
5. Проведено тестування взаємодії компонентів системи.

РОЗДІЛ 4. РОЗРОБКА СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї проекту

Незважаючи на те, що технології доповненої реальності широко представлені на ринку товарів і послуг, створення сервісу, що об'єднує в собі функції інструменту доповненої реальності і маркетплейсу для розміщення моделей, є необхідністю. У розділі 1 було розглянуто існуючі рішення систем доповненої реальності, що відповідають базовим вимогам.

В ході розробки проекту, була представлена ідея для реалізації: мультиплатформенний мобільний додаток з технологією доповненої реальності і веб-додаток для менеджменту моделей. Важливим доповненням є реалізація функції маркетплейса всередині як мобільного додатку, так і веб-сервісу. Розроблений продукт не може бути представлений серед сервісів доповненої реальності без розробки маркетингових стратегій та оцінки конкурентоспроможності [15]. У таблиці 4.1 представлені розвитки рішень основних параметрів системи, що дозволяють ефективно реалізувати ідею проекту.

Таблиця 4.1 - Можливі варіанти рішень – засоби реалізації кожної функції

Основні параметри	Проміжні рішення				
	1-ше	2-ше	3-ше	4-ше	5-ше
Пристрої виводу	ПК	ПК	Шолом віртуальної /доповненої реальності	Мобільний пристрій	Мобільний пристрій

Технології візуалізації	Без захоплення	Без захоплення	Захоплення поверхні	Захоплення маркерів	Захоплення поверхні і маркерів
Фреймворк	Glide	SceneForm	SceneForm	SceneKit	SceneKit
GLTF-підтримка	-	-	-	+	+

Таким чином, ідею проекту можна сформулювати через наступні оптимальні та інноваційні значення:

- підтримка на мобільних пристроях різних типів з різними операційними системами;
- можливість захоплення як поверхні, так і маркерів;
- використання технологій ARKit та ARCore.

4.1.1 Задум проекту

- 1. Проект за задумом – система відтворює зображення на поверхню, з огляду на масштаб і текстури. Дозволяє користувачеві використовувати багатофункціональні моделі для реалізації завдань.
- 2. Проект у реальному виконанні – система використовує технології доповненої реальності для основних операційних систем, поглиблюється в реалістичне відтворення масштабованих моделей, дозволяє реалізувати власні моделі незалежно від їх типу. Продукт і його вартість визначаються кількістю функціоналу, запитаним користувачем.

- 3. Проект з підкріпленням – система розширює базу моделей і пропонує додаткову багатокористувацьку структуру для оптимізованого коригування моделей.

4.1.2 Удосконалення продукту

Таблиця 4.2 - Опрацювання питань для удосконалення продукту

№ з/п	Запитання	Відповідь
1	2	3
1	Частиною яких систем є продукт?	AR-додаток доповненої реальності. Є частиною ІТ-ринку.
2	Які функції надсистеми може виконувати продукт? Як їх з ним пов'язати?	Може додатково замінювати дизайнерів і проектувальників при створенні макетів.
3	Чи можна розділити продукт на частини?	Ні
4	Чи можна об'єднати (агрегувати) кілька елементів продукту в один?	Ні
5	Чи можна нерухомі частини продукту зробити рухомими і навпаки?	Ні

6	Яким має бути ідеальний продукт?	Без використання громіздких пристроїв відтворення.
7	Що відбудеться, якщо вилучити цей продукт? Чим його можна замінити?	На українському ринку поки немає заміни даного продукту.
8	Яким цей продукт був у минулому?	Вивід додаткових елементів на екран ПК без захоплення поверхні і масштабування.
9	На розвиток яких функцій було спрямоване удосконалення продукту?	Розвивалися функції трекінгу. Тепер можливе захоплення обличчя, поверхонь, маркерів- об'єктів, окремих зображень.
10	Які функції залишилися «недорозвиненими»?	Функція підтримки однієї сесії з різних пристроїв.
11	Як можна натеper розвинути ці функції?	Створити мультикористувацьку платформу для перегляду моделей.

4.2 Аналіз ринкових можливостей запуску стартап-проекту

Таблиця 4.3 - Попередня характеристика потенційного ринку стартап-проекту

п/п	Показники стану ринку (найменування)	Характеристика
	Кількість головних гравців, од	3
	Загальний обсяг продаж, грн/ум.од	10 000 000
	Динаміка ринку (якісна оцінка)	стагнує
	Наявність обмежень для входу (вказати характер обмежень)	Відсутні через низьку конкуренцію на внутрішньому ринку
	Специфічні вимоги до стандартизації та сертифікації	Відповідають всім стандартам
	Середня норма рентабельності в галузі (або по ринку), %	60

1. ROAR та Augment – потенційні конкуренти
2. $(600 \cdot 500 + 1000 \cdot 100) \cdot 25 = 10\,000\,000$
3. Немає компанії, яка б просувала ідею використання AR в різних сферах життєдіяльності
4. Низька конкуренція на внутрішньому ринку
5. Сервіс відповідає стандартам оформлення мобільних додатків та веб-сайтів відповідно до технічних та юридичних вимог онлайн-магазинів додатків (Google Play, App Store)
6. Середня норма рентабельності по ринку вираховується як відношення середнього річного доходу до вартості першочергових інвестицій $= 300\,000 / 500\,000 = 60\%$ [16].

Таблиця 4.4 - Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінно сті у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	Візуаліза ція товарів та винаходів	Архітекту рні бюро, рекламні агенції	Різні концепти впровадження та просування товарів	Інтерактивні сть, якість дизайну

Таблиця 4.5 - Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
	Конкуренція в сегменті	Поява нових компаній з додатками доповненої реальності	Підсилення рекламної кампанії
	Висока ціна	Користувач не бажає купляти преміум-версії	Акції та розпродажі

Таблиця 4.6 - Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
	Розширення бази даних	Створення більшої кількості готових моделей для залучення нових клієнтів	Залучення великого штату дизайнерів та розширення виробництва
	Інноваційність	Продукт не має аналогів та прямих конкурентів на ринку	Розвиток додаткових функцій та налаштувань

Таблиця 4.7 - Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції - монополія	До запуску продукту компанія ще не має конкурентів на внутрішньому ринку, що дає велику перевагу.	Первісний контроль сфери
2. За рівнем конкурентної боротьби - міжнародний	Продукт може бути використаний в системах різних країн	Розвиток міжнародних зв'язків

Продовження таблиці 4.7

3. За галузевою ознакою - міжгалузева	Архітектура, бізнес, освіта, реклама, тощо	Є можливості подальшого розширення до інших галузей
4. Конкуренція за видами товарів: - товарно-видова	Між продуктами зі схожими функціями	Підвищити якість товару, зниження ціни
5. За характером конкурентних переваг - цінова	Міжнародні конкуренти мають вищу ціну на продукт	Вихід на міжнародний ринок
6. За інтенсивністю - не марочна	Марка не є значною	Реклама бренду

Таблиця 4.8 - Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Поста-чальники	Клієнти	Товари-замінники
	IKEA Place	-	Кількість	Інтерактивність	Вартість бренду

Висновки:	Конкурент не розглядає можливість працювати в різних галузях	Ні	Ні, оскільки продукт є інноваційним	Так, клієнти визначають вигляд продукту	Через велику кількість фізичних точок продажу та використання, конкурент має вагому перевагу.
-----------	--	----	-------------------------------------	---	---

Таблиця 4.9 - Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
	Ціна	Ціна нижча за ціни продуктів з аналогічними функціями іноземних компаній

	Інтерактивність	Користувач може керувати моделлю за допомогою маркерів
	Універсальність	Можна створити чи завантажити будь-яку модель(форма, колір, текстура тощо)
	Доступність	Користувачу пропонується вибрати найбільш зручну для нього версію підписки

Таблиця 4.10 - Порівняльний аналіз сильних та слабких сторін

п/ п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з AR						
			3	2	1		1	2	3
	Ціна	15							
	Інтерактивність	17							
	Універсальність	19							
	Доступність	18							

Таблиця 4.11 - SWOT- аналіз стартап-проекту

Сильні сторони: Залученність у різні галузі Інноваційність [17]	Слабкі сторони: Непопулярність Маловідомість
Можливості: Вихід на міжнародний ринок Розширення продажів в інших галузях	Загрози: Поява прямих конкурентів

Таблиця 4.12 - Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
	Перехід до інших галузей реалізації	Висока	6 місяців
	Пошук нових компаній	Середня	1 рік

4.3 Розроблення ринкової стратегії проекту

Таблиця 4.13 - Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенцій- них клієнтів	Готовн ість споживачів сприйняти продукт	Орієнто вний попит в межах цільової групи (сегменту)	Іnten сив-ність конку- ренції в сегменті	Прост ота входу у сегмент
	Архітек тур-ні бюро	Висока зацікавлені сть (для візуалізації макетів)	Високий – через інновацій- ність	низьк а	Не має конкурент ів серед архітектур -них додатків

Продовження таблиці 4.13

	Університети та школи	Висока зацікавленість (для інтерактивності навчального процесу)	Висока й - через інноваційність	середня	Є конкуренти з додатками-аналогами, але вужчим функціоналом
	Рекламні агенції	Висока зацікавленість (для використання у інтернет-сегменті)	Висока й – через велику конкуренцію в рекламному сегменті	середня	Не має конкурентів серед універсальних додатків, але існують власні напрацювання рекламної сфери
Які цільові групи обрано: архітектурні бюро, університети та школи, рекламні агенції					

Таблиця 4.14 - Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні і позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Залучення великого штату спеціалістів	Робить ставку на додаткових можливостях виробництва нових продуктів.	Існуюче виробництво є центром бізнесу, можливе розширення асортименту.	Стратегія центрованої диверсифікації

Таблиця 4.15 - Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем » на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів ?	Чи буде компанія копіювати основні характеристик и товару конкурента, і які?	Стратегія конкурентної поведінки*
	Продукція є першопрохідцем на ринку.	Буде шукати нових споживачів.	Компанія не буде копіювати характеристик и товарів закордонних конкурентів.	Ринкова активність, захоплення незайнятих просторів

Таблиця 4.16 - Визначення стратегії позиціонування

п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформулювати комплексну позицію власного проекту (три ключових)
	Висока продуктивність виробу, універсальність системи	Стратегія центральної диверсифікації	Існуюче виробництво є центром бізнесу, можливе розширення асортименту.	Висока якість, індивідуальний підхід, довгострокова технічна підтримка, доступність.

4.4 Розроблення маркетингової програми стартап-проекту

Таблиця 4.17 - Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба [18]	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Продуктивність	Забезпечує оптимізацію продукту на сервері	Зміна продукту розробником можлива в будь-який момент на будь-якому етапі виробництва

Продовження таблиці 4.17

2	Деталізація	Можливість вибору з бази моделей, що постійно доповнюється та модифікується	Великий асортимент доступних моделей і можливість замовлення власної моделі для розробки
3	Універсальність	Система дозволяє розмістити модель в усіх трьох режимах – підлога, стіна та захоплення маркерів.	Універсальність системи.

Таблиця 4.18 - Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за задумом	Залучити користувачів до використання доповненої реальності в різних сферах життєдіяльності

II. Товар у реальному виконанні	Властивості/характеристики	М/Н м	Вр/Тх /Тл/Е/Ор
	1. World Tracking (трекінг навколишнього середовища)	Нм	Тх
	2. Face Tracking (трекінг обличчя)	Нм	Тх
		Нм	Тх
	3. Object Tracking (трекінг об'єктів)	Нм	Тх
		Нм	Тх
	4. Image Tracking (трекінг зображень)	Нм	Тх
	5. Orientation Tracking (орієнтаційний трекінг)	Нм	Тх/Ор
	6. Positional Tracking (позиційний трекінг)	М	Тх/Тл
	7. Ефекти візуалізації		
	8. Багатокористувацькість		
Якість: патенти на використання AR-технологій, захист авторських прав власників розроблених моделей			
Дизайн/стиль: розробка зручного для користувача інтерфейсу для інтеграції моделей в мобільні операційні системи			
III. Товар із підкріплен ням	До продажу: можливість редагування моделей в проектах.		
	Після продажу: можливість створення модифікації існуючих моделей та додавання нових до бази.		
За рахунок чого потенційний товар буде захищено від копіювання: відповідно до державного та правового законодавства, за рахунок патентів.			

Таблиця 4.19 - Визначення меж встановлення ціни

№ п/п	Рівень цін на товари- замінники	Рівень цін на товари- аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	Ціна товару базується на ціні пристрою з операційною системою, що підтримує товар- замінник (\$50- 100/міс, \$500- 1000/рік)	Ціна товару базується на ціні пристрою з операційною системою, що підтримує товар-аналог (\$30- 100/міс, \$200- 700/рік)	\$7000- 15000/рік – дизайнери \$30000- 100000/рік - компанії	Безкоштовно (мінімальна версія з вільним доступом) - 600\$/рік

Таблиця 4.20 - Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
----------	---	---	-------------------------	-----------------------------

1	Отримання довгострокової підписки	Встановлення контактів із споживачами. Підтримка при навчанні у використанні. Формування попиту. Аналіз попиту на існуючі базові моделі та на створення/замовлення нових.	0 (без посередників)	Пряма поставка товару в компанії, які купують товар для реklamних цілей, в архітектурні бюро, в університети.
---	---	---	-------------------------	---

Таблиця 4.21.1 - Концепція маркетингових комунікацій (1)

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування
Отримання довгострокової підписки	Формальні/ неформальні канали комунікацій	Доступність; Універсальність; Можливість оптимізації.

Таблиця 4.21.2 - Концепція маркетингових комунікацій (2)

Специфіка поведінки цільових клієнтів	Завдання рекламного повідомлення	Концепція рекламного звернення
Отримання довгострокової підписки	Інформування споживачів; Інформування в засобах масової інформації Пошук дизайнерів; Пошук інвесторів.	Даний продукт є інноваційним, інтелектуальним, естетично- та візуально-цінним.

ВИСНОВОК ДО РОЗДІЛУ 4

У Розділі 4 цієї дипломної роботи був розроблений стартап-проект, який необхідний для реалізації готової системи доповненої реальності з функцією маркетплейсу.

Описана ідея проекту та визначено основні інноваційні рішення розробленої системи. Також були опрацьовані варіанти для удосконалення проекту.

В ході аналізу ринковий можливостей запуску стартап-проекту:

- 1) складена характеристика потенційного ринку стартап-проекту за основними показниками стану ринку;
- 2) за потребами, які формують ринок, складена характеристика потенційних клієнтів;
- 3) проведена оцінка конкуренції в галузі;
- 4) проведений SWOT-аналіз проекту.

Також були представлені альтернативи ринкового впровадження стартап-проекту.

При розробці ринкової стратегії проекту:

- 1) визначені базові стратегії розвитку проекту та конкурентної поведінки;
- 2) обрані цільові групи потенційних клієнтів.

В ході розробки маркетингової програми:

- 1) проаналізовані ключові переваги концепції продукту;
- 2) описані рівні моделі товару;
- 3) сформована система збуту продукту і концепція маркетингової комунікації.

На основі виконаної роботи був отриманий стартап-проект для системи доповненої реальності з функцією маркетплейсу, що дозволяє продукту повноцінно вийти на ринок послуг.

ВИСНОВОК

Даний магістерський проект присвячений системі доповненої реальності з функцією маркетплейсу. В ході виконання роботи був вивчений процес синтезу технологій доповненої реальності з можливими стратегіями впровадження елементів маркетплейсу в систему.

В розділі 1 були вивчені існуючі рішення в основних сферах, в яких використовуються технології AR, проведено аналіз переваг і недоліків різних додатків і визначені необхідні рішення для усунення поточних проблем.

У розділі 2 була вивчена предметна область завдання, були визначені основні вимоги та функції системи, визначена структура та наголошено на необхідності наявності веб-додатку для зручності завантаження моделей. Виходячи з цих вимог і функцій, були сформовані основні сценарії роботи сервісу та інтерфейс системи.

У Розділі 3 був проведений порівняльний аналіз технологій, які використовуються для відображення моделей в доповненої реальності. Були обрані найбільш оптимальні технології для реалізації компонентів системи.

В результаті роботи над системою було реалізовано серверну частину і 3 клієнтських додатки, які виконують функції, заявлені в другому розділі. Також було проведено тестування взаємодії компонентів системи і проведена оптимізація, для поліпшення UX були додані підказки і скоректований дизайн клієнтських додатків.

Для створення повноцінного конкурентоспроможного продукту був створений стартап-проект, де було проаналізовано основні фактори можливостей і ризиків впровадження системи на ринок послуг, сформовані ринкові і маркетингові стратегії.

Результат проведеної роботи - система доповненої реальності з функцією маркетплейсу, яка без використання додаткових ресурсів дозволяє як користувачу отримати зручний і простий у використанні додаток для

перегляду моделей в доповненій реальності, так і забезпечити повноцінну взаємодію між розробниками і клієнтами.

Список використаної літератури

1. M. Kesim^a, Y. Ozarslan. Augmented Reality in Education: Current Technologies and the Potential for Education / Mehmet Kesim^a, Yasin Ozarslan // Procedia - Social and Behavioral Sciences. – 2012. – Volume 47. – pages 297-302
2. T. Khan, K. Johnston, J. Ophoff. The Impact of an Augmented Reality Application on Learning Motivation of Students/ Tasneem Khan, Kevin Johnston, Jacques Ophoff // Advances in Human-Computer Interaction. – 2019. – Volume 2019.
3. Shuai Yang, Sixing Chen. How augmented reality affects advertising effectiveness: The mediating effects of curiosity and attention toward the ad / Shuai Yang, Sixing Chen // Journal of Retailing and Consumer Services. – 2020. – Volume 54.
4. Practical Augmented Reality: A Guide to the Technologies, Applications, and Human Factors for AR and VR (Usability)/ Steve Aukstakalnis, 2016.- 448 с.
5. C# docs - get started, tutorials, reference. | Microsoft Docs [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/dotnet/csharp/>.
6. Documentation - Python.org [Електронний ресурс] – Режим доступу до ресурсу: <https://www.python.org/doc/>.
7. MySQL Documentation – MySQL [Електронний ресурс] – Режим доступу до ресурсу: <https://dev.mysql.com/doc/>.
8. Manuals – PostgreSQL [Електронний ресурс] – Режим доступу до ресурсу: <https://www.postgresql.org/docs/manuals/>.
9. ARKit | Apple Developer Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.apple.com/documentation/arkit>.
10. ARCore API reference | Google Developers [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.google.com/ar/reference>.

11. Android development documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.android.com/index.html>.
12. Vuforia Library – Vuforia [Електронний ресурс] – Режим доступу до ресурсу: <https://library.vuforia.com/getting-started/overview.html>.
13. React Native [Електронний ресурс] – Режим доступу до ресурсу: <https://github.io/react-native/>.
14. Руководство по Angular – Metanit [Електронний ресурс] – Режим доступу до ресурсу: <https://metanit.com/web/angular2/>.
15. Менеджмент стартап-проектів: Навчально-методичний комплекс дисципліни [Електронний ресурс]: навч. посіб. для студ. спеціальності 073 «Менеджмент» / КПІ ім. Ігоря Сікорського; уклад.: К. О. Бояринова. – Електронні текстові дані (1 файл: 5,85 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2020. – 153 с.
16. Gans J, Erin L. Scott, Stern S. Strategy for Start-ups [Електронний ресурс] – Режим доступу до ресурсу: <https://hbr.org/2018/05/do-entrepreneurs-need-a-strategy#strategy-for-start-ups>.
17. SWOT анализ: методика определения угроз и возможностей. Онлайн — энциклопедия по маркетингу и рекламе. [Електронний ресурс] – Режим доступу до ресурсу: <http://powerbranding.ru/biznes-analiz/swot/ugrozy-vozmozhnosti/>.
18. Мулик Л. І., Солнцев С. О. Маркетинг незалежних та афілійованих стартапів. Маркетинг незалежних та афілійованих стартапів. Економічний вісник Національного технічного університету України «Київський політехнічний інститут». 2017. № 14. С.336-342.

ДОДАТКИ

ДОДАТОК А

ЛІСТИНГ

```
using ARenaWebClient.Helpers;
using ARenaWebClient.Services;
using Infrastructure.Helpers;
using Infrastructure.Models;
using Microsoft.AspNetCore.Authentication.JwtBearer;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Http.Features;
using Microsoft.AspNetCore.HttpsPolicy;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.SpaServices.AngularCli;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;

namespace ARenaWebClient
{
    public class Startup
    {
        public Startup(IConfiguration configuration)
        {
            Configuration = configuration;
        }

        public IConfiguration Configuration { get; }

        // This method gets called by the runtime. Use this method to add services to the container.
        public void ConfigureServices(IServiceCollection services)
        {
            services.AddDbContext<ARenaContext>(options =>
options.UseNpgsql(Configuration.GetConnectionString("DefaultConnection")));

            services.AddIdentity<ApplicationUser, IdentityRole>().AddEntityFrameworkStores<ARenaContext>();
            services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme).AddJwtBearer();
        }
    }
}
```

```

services.AddScoped<IBackendRepository, BackendRepository>();
services.AddScoped<ICompanyService, CompanyService>();
services.AddScoped<IProductsService, ProductsService>();
services.AddScoped<IMemoryService, MemoryService>();

services.Configure<IdentityOptions>(options =>
{
    // Password settings
    options.Password.RequireDigit = false;
    options.Password.RequiredLength = 6;
    options.Password.RequireNonAlphanumeric = false;
    options.Password.RequireUppercase = false;
    options.Password.RequireLowercase = false;
});

services.Configure<FormOptions>(options =>
{
    options.MultipartBodyLengthLimit = 6000000000;
});

services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_1);

// In production, the Angular files will be served from this directory
services.AddSpaStaticFiles(configuration =>
{
    configuration.RootPath = "ClientApp/dist";
});
}

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
}

```

```

else
{
    app.UseExceptionHandler("/Error");
    app.UseHsts();
}

app.UseStaticFiles();
app.UseSpaStaticFiles();

app.UseAuthentication();

app.UseMvc(routes =>
{
    routes.MapRoute(
        name: "default",
        template: "{controller}/{action=Index}/{id?}");
});

app.UseSpa(spa =>
{
    // To learn more about options for serving an Angular SPA from ASP.NET Core,
    // see https://go.microsoft.com/fwlink/?linkid=864501

    spa.Options.SourcePath = "ClientApp";

    if (env.IsDevelopment())
    {
        spa.UseAngularCliServer(npmScript: "start");
    }
});

}

}

}

using ARenaWebClient.Helpers;
using AutoMapper;
using Infrastructure.Models;
using Microsoft.AspNetCore.Authorization;

```

```

using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.IdentityModel.Tokens;
using System.IdentityModel.Tokens.Jwt;
using System.Security.Claims;

namespace ARenaWebClient.Controllers
{
    [AllowAnonymous]
    [Route("api/[controller]")]
    public class AuthorizeController: Controller
    {
        private readonly UserManager<ApplicationUser> _userManager;
        private readonly SignInManager<ApplicationUser> _signInManager;
        private readonly IBackendRepository _repository;

        public AuthorizeController(UserManager<ApplicationUser> userManager, SignInManager<ApplicationUser>
signInManager, IBackendRepository repository)
        {
            this._userManager = userManager;
            this._signInManager = signInManager;
            this._repository = repository;
        }

        [HttpPost("[action]")]
        public async Task<ActionResult> Login([FromBody]LoginViewModel model)
        {
            try
            {
                if (ModelState.IsValid)
                {
                    var result = await _signInManager.PasswordSignInAsync(model.Email, model.Password,
model.RememberMe, false);
                    if (result.Succeeded)

```

```

{
    var identity = await GetIdentity(model.Email);
    if (identity == null)
    {
        Response.StatusCode = 400;
        return BadRequest("IncorrectNameOrPassword");
    }

    var now = DateTime.UtcNow;
    // создаем JWT-токен
    var jwt = new JwtSecurityToken(
        //issuer: AuthOptions.ISSUER,
        //audience: AuthOptions.AUDIENCE,
        claims: identity.Claims,
        expires: now.Add(TimeSpan.FromMinutes(30))
        //signingCredentials: new SigningCredentials(AuthOptions.GetSymmetricSecurityKey(),
SecurityAlgorithms.HmacSha256)
    );
    var encodedJwt = new JwtSecurityTokenHandler().WriteToken(jwt);

    var response = new
    {
        access_token = encodedJwt,
        username = identity.Name
    };

    // сериализация ответа
    return Json(response);
}
else
{
    return Unauthorized();
}
}
else
{
    return Unauthorized();
}
}

```

```
    }  
    catch (Exception ex)  
    {  
        return BadRequest();  
    }  
}
```

```
[HttpPost("{action}")]  
public async Task<IActionResult> Logout()  
{  
    try  
    {  
        await _signInManager.SignOutAsync();  
        return Ok();  
    }  
    catch (Exception ex)  
    {  
        return BadRequest();  
    }  
}
```

```
[HttpPost("{action}")]  
public async Task<IActionResult> Register([FromBody]RegistrationViewModel model)  
{  
    try  
    {  
        if (!ModelState.IsValid)  
        {  
            return BadRequest(ModelState);  
        }  
  
        ApplicationUser user = new ApplicationUser(model.Username, model.Email);  
        var result = await _userManager.CreateAsync(user, model.Password);  
  
        if (!result.Succeeded)  
        {  
            return BadRequest(result.Errors);  
        }  
    }  
}
```



```

return Json("Account created");
    }
    catch (Exception ex)
    {
        return BadRequest(ex);
    }
}

private async Task<ClaimsIdentity> GetIdentity(string email)
{
    ApplicationUser person = await _userManager.FindByEmailAsync(email);
    if (person != null)
    {
        var claims = new List<Claim>
        {
            new Claim(ClaimsIdentity.DefaultNameClaimType, person.UserName)
        };
        ClaimsIdentity claimsIdentity =
            new ClaimsIdentity(claims, "Token", ClaimsIdentity.DefaultNameClaimType,
                ClaimsIdentity.DefaultRoleClaimType);
        return claimsIdentity;
    }

    // если пользователя не найдено
    return null;
}

}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using ARenaWebClient.Helpers;
using ARenaWebClient.Services;
using Infrastructure.Models;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;

```

```

namespace ARenaWebClient.Controllers
{
    [Route("api/[controller]")]
    [Authorize]
    public class ProductsController : Controller
    {
        private IProductsService _service;
        private readonly UserManager<ApplicationUser> _userManager;
        private Task<ApplicationUser> GetCurrentUserAsync() => _userManager.GetUserAsync(HttpContext.User);

        public ProductsController(UserManager<ApplicationUser> userManager)
        {
            _userManager = userManager;
            _service = new ProductsService(new ModelStateWrapper(ModelState), new BackendRepository(new
ARenaContext()));
        }

        [HttpGet("[action]")]
        public async Task<ResponseData<List<Product>>> GetProducts()
        {
            try
            {
                ApplicationUser user = await GetCurrentUserAsync();
                return new ResponseData<List<Product>>>(await _service.GetProducts(user.Id));
            }
            catch(Exception ex)
            {
                return new ResponseData<List<Product>>>(new List<Product>(), ex.Message);
            }
        }

        [HttpGet("[action]")]
        public async Task<ResponseData<Product>> GetProduct(int productId)
        {
            try
            {

```

```

        return new ResponseData<Product>(await _service.GetProduct(productId));
    }
    catch (Exception ex)
    {
        return new ResponseData<Product>(null, ex.Message);
    }
}

[HttpGet("[action]")]
public async Task<ResponseData<List<Product>>> GetCompaniesProducts(int companyId)
{
    try
    {
        return new ResponseData<List<Product>>(await _service.GetCompanyProducts(companyId));
    }
    catch (Exception ex)
    {
        return new ResponseData<List<Product>>(new List<Product>(), ex.Message);
    }
}

[HttpPost("[action]")]
public async Task<ResponseData<Product>> AddProduct([FromBody] Product product)
{
    try
    {
        return new ResponseData<Product>(await _service.AddProduct(product));
    }
    catch (Exception ex)
    {
        return new ResponseData<Product>(null, ex.Message);
    }
}

[HttpDelete("[action]")]
public async Task<ResponseData<int>> RemoveProduct(int id)
{
    try

```

```

        {
            return new ResponseData<int>(await _service.RemoveProduct(id));
        }
        catch (Exception ex)
        {
            return new ResponseData<int>(-1, ex.Message);
        }
    }

    [HttpPut("[action]")]
    public async Task<ResponseData<int>> RestoreProduct([FromBody]int id)
    {
        try
        {
            return new ResponseData<int>(await _service.RestoreProduct(id));
        }
        catch (Exception ex)
        {
            return new ResponseData<int>(-1, ex.Message);
        }
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using ARenaWebApi.Controllers;
using Infrastructure.Models;
using static ARenaWebApi.Controllers.ApiController;

namespace ARenaWebApi.Helpers
{
    public class ArenaRepository : IArenaRepository
    {
        private ARenaContext context;

        public ArenaRepository(ARenaContext context)
        {

```

```
this.context = context;
}

public List<CompanyType> GetCompanyTypes()
{
    return (from categoryType in context.CompanyTypes
        orderby categoryType.ID
        select new CompanyType
        {
            ID = categoryType.ID,
            Name = categoryType.Name,
            Description = categoryType.Description,
            MainCategories = context.LibCategories.Where(x => x.CategoryTypeID == categoryType.ID).ToList()
        }).ToList();
}

public List<CompanyGetModel> GetCompanies(CompaniesGetParams data)
{
    var companies = (from company in context.Companies
        orderby company.ID
        where String.IsNullOrEmpty(data.Name) || company.Name.ToLower().Contains(data.Name.ToLower())
        where ((data.CompanyTypeIDs.Length == 0) || data.CompanyTypeIDs.Contains(company.CompanyTypeID))
        where company.Status == Status.Active
        select new CompanyGetModel
        {
            ID = company.ID,
            Name = company.Name,
            SiteLink = company.SiteLink,
            CompanyTypeID = company.CompanyTypeID,
            Settings = context.CompanySettings.Where(x => x.CompanyID == company.ID).ToList(),
            MainCategories = (from libCategory in context.LibCategories
                join product in context.Products on libCategory.ID equals product.LibCategoryId
                where product.LibCategoryId == libCategory.ID && product.CompanyID == company.ID
                select new MainCategoryModel
                {
                    CompanyID = company.ID,
                    Name = libCategory.Name,
```

```

        LibCategoryID = libCategory.ID,
        ID = libCategory.ID,
        CustomCategories = context.CustomCategories.Where(x => x.CompanyID ==
company.ID && x.CategoryID == libCategory.ID).ToList()
    }).GroupBy(x => x.ID).Select(x => x.First()).ToList()
    });
    if (data.CategoryIDs.Length != 0)
    {
        var companyIds = (from product in context.Products
            where data.CategoryIDs.Contains(product.LibCategoryID)
            select product.CompanyID).ToList();
        companies = companies.Where(x => companyIds.Contains(x.ID));
    }
    return companies.ToList();
}

public List<Product> GetProducts(ProductsGetParams data)
{
    return (from product in context.Products
        orderby product.ID
        join customcategory in context.CustomCategories on product.CustomCategoryID equals customcategory.ID
        where String.IsNullOrEmpty(data.Name) || product.Name.ToLower().Contains(data.Name.ToLower())
        where data.CompanyID == customcategory.CompanyID
        where data.CustomCategoriesIDs.Length == 0 ||
data.CustomCategoriesIDs.Contains(product.CustomCategoryID)
        where product.Status == Status.Active
        select new Product
        {
            ID = product.ID,
            Name = product.Name,
            CustomCategoryID = product.CustomCategoryID,
            ArScanType = product.ArScanType,
            CompanyID = product.CompanyID,
            LibCategoryID = product.LibCategoryID,
            Status = product.Status,
            MarkerImages = product.MarkerImages,
            ModelUnit = product.ModelUnit,
            ContactID = product.ContactID,
            Description = product.Description,
            Images = product.Images,

```

```

        ModelFileName = product.ModelFileName,
        ProductCode = product.ProductCode,
        Contact = context.Contacts.Where(x => x.ProductID == product.ID).FirstOrDefault()
    }).ToList();
}

public int GetCompaniesCount()
{
    return context.Companies.Where(x => x.Status == Status.Active).Count();
}

public SearchCompanyResult SearchCompanies(string query, int limit, int[] companyTypeIds, int[] categoriesIds)
{
    query = query.ToLower();
    using (context)
    {
        var res = (from company in context.Companies
                    orderby company.ID
                    where company.Name.ToLower().Contains(query)
                    where company.Status == Status.Active
                    && (categoriesIds.Count() == 0 || categoriesIds.Intersect(
                        (from product in context.Products
                         where product.CompanyID == company.ID
                         select product.LibCategoryID).ToArray()).Any())
                    where companyTypeIds.Contains(company.CompanyTypeID)
                    select new SearchCompanyItem {
                        CompanyID = company.ID,
                        CompanyName = company.Name
                    });

        return new SearchCompanyResult()
        {
            CompaniesCount = res.Count(),
            FoundCompanies = res.Take(limit).ToList()
        };
    }
}

```

```

public SearchProductResult SearchProducts(string query, int companyId, int limit, int[] categoriesIds)
{
    query = query.ToLower();
    using (context)
    {
        var res = (from product in context.Products
                    orderby product.ID
                    where product.Status == Status.Active
                    where product.Name.ToLower().Contains(query)
                    && product.CompanyID == companyId
                    && (categoriesIds.Count() == 0 || categoriesIds.Contains(product.CustomCategoryID))
                    select product.Name);

        return new SearchProductResult()
        {
            ProductsCount = res.Count(),
            FoundProducts = res.Take(limit).ToList()
        };
    }
}

public List<RecentGetModel> GetRecent(RecentGetParams data)
{
    return (from product in context.Products
            where product.Status == Status.Active
            orderby product.ID descending
            select new RecentGetModel
            {
                ID = product.ID,
                Name = product.Name,
                CompanyID = product.CompanyID,
                LibCategoryID = product.LibCategoryID,
                CustomCategoryID = product.CustomCategoryID,
                ContactID = product.ContactID,
                Description = product.Description,
                Images = product.Images,
                ModelFileName = product.ModelFileName,
                ProductCode = product.ProductCode,
                ArScanType = product.ArScanType,
            }
    );
}

```



```

        MarkerImages = product.MarkerImages,
        ModelUnit = product.ModelUnit,
        Status = product.Status,
        Contact = context.Contacts.Where(x => x.ProductID == product.ID).FirstOrDefault(),
        CompanyName = context.Companies.Where(x => x.ID == product.CompanyID).FirstOrDefault().Name
    }).Take(data.Limit).ToList();
}

public async Task<bool> AddStatistics(Statistic data)
{
    try
    {
        if(data.ID != 0 || data.ModelID == null || data.ModelID <= 0)
        {
            throw new InvalidOperationException();
        }

        data.ID = context.Statistics.Max(x => x.ID) + 1;

        await context.Statistics.AddAsync(data);

        await context.SaveChangesAsync();

        return true;
    }
    catch (Exception ex)
    {
        return false;
    }
}

}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Infrastructure.Models;
using Microsoft.EntityFrameworkCore;
using static ARenaWebClient.Controllers.CompaniesController;

namespace ARenaWebClient.Helpers

```

```

{
    public class BackendRepository : IBackendRepository
    {
        private ARenaContext context;

        public BackendRepository(ARenaContext context)
        {
            this.context = context;
        }

        public async Task<List<Company>> GetCompanies(string userId)
        {
            using (context)
            {
                return await context.Companies.Where(x => x.UserID == userId).ToListAsync();
            }
        }

        public async Task<List<Product>> GetProducts(string userId)
        {
            using (context)
            {
                return await (from company in context.Companies
                              where company.UserID == userId
                              join product in context.Products on company.ID equals product.CompanyID
                              select product).ToListAsync();
            }
        }

        public async Task<Company> AddCompany(AddCompanyModel model)
        {
            using (context)
            {
                int lastCompanyId = context.Companies.Max(x => x.ID);
                model.company.ID = lastCompanyId + 1;
                model.company.Status = Status.Active;
                context.Companies.Add(model.company);
                await context.SaveChangesAsync();

                Company addedCompany = context.Companies.Find(model.company.ID);
                int lastSettingId = context.CompanySettings.Max(x => x.ID) + 1;
            }
        }
    }
}

```

```

        foreach (CompanySetting setting in model.settings)
        {
            setting.ID = lastSettingId;

            setting.CompanyID = addedCompany.ID;

            context.CompanySettings.Add(setting);

            lastSettingId++;
        }

        await context.SaveChangesAsync();

        return addedCompany;
    }
}

public async Task<List<CompanyType>> GetCompanyTypes()
{
    using (context)
    {
        return await context.CompanyTypes.ToListAsync();
    }
}

public async Task<List<LibCategory>> GetMainCategories()
{
    using (context)
    {
        return await context.LibCategories.ToListAsync();
    }
}

public async Task<List<CustomCategory>> GetCustomCategories(int companyId)
{
    using (context)
    {
        return await context.CustomCategories.Where(x => x.CompanyID == companyId).ToListAsync();
    }
}

public async Task<List<CustomCategory>> GetCustomCategories(int companyId, int categoryId)
{

```

```

        using (context)
        {
            return await context.CustomCategories.Where(x => x.CompanyID == companyId && x.CategoryID ==
categoryId).ToListAsync();
        }
    }
}

```

```

public async Task<List<Product>> GetCompanyProducts(int companyId)
{
    using (context)
    {
        return await context.Products.Where(x => x.CompanyID == companyId).ToListAsync();
    }
}

public async Task<Product> AddProduct(Product product)
{
    using (context)
    {
        //Если БД сгенерирована с Seed данными, то автоинкремент для ID не работает
        int contactId = context.Contacts.Max(x => x.ID) + 1;
        int productId = context.Products.Max(x => x.ID) + 1;
        Contact contact = new Contact() {
            ID = contactId,
            Email = "test@ukr.net",
            Phone = "380971232323",
            ProductID = productId,
            ProductLink = "https://google.com",
            Telegram = "@companyName",
            Viber = "0975614482",
            Instagram = "@companyName",
        };
        contact.ID = context.Contacts.Max(x => x.ID) + 1;
        context.Contacts.Add(contact);
        product.Contact = contact;
        product.ID = productId;
        product.ContactID = contact.ID;
        product.Status = Status.Active;
        context.Products.Add(product);
    }
}

```

```

        await context.SaveChangesAsync();

        return product;
    }
}

public async Task<Product> GetProduct(int productId)
{
    using (context)
    {
        return await context.Products.FindAsync(productId);
    }
}

public async Task<Company> GetCompany(int companyId)
{
    using (context)
    {
        return await context.Companies.FindAsync(companyId);
    }
}

public async Task<CustomCategory> AddCustomCategory(CustomCategory category)
{
    int categoryId = context.CustomCategories.Max(x => x.ID) + 1;
    category.ID = categoryId;
    context.CustomCategories.Add(category);
    await context.SaveChangesAsync();
    return category;
}

public async Task<int> RemoveProduct(int productId)
{
    try
    {
        using (context) {
            Product currProduct = context.Products.Find(productId);
            currProduct.Status = Status.Blocked;
            int res = await context.SaveChangesAsync();

```

```

        return res;
    }
}
catch(Exception ex)
{
    return -1;
}
}

```

```

public async Task<int> RemoveCompany(int companyId)
{
    try
    {
        using (context)
        {
            Company currCompany = context.Companies.Find(companyId);
            currCompany.Status = Status.Blocked;
            var companyProducts = context.Products.Where(x => x.CompanyID == companyId);
            foreach(Product product in companyProducts)
            {
                product.Status = Status.Blocked;
            }
            int res = await context.SaveChangesAsync();
            return res;
        }
    }
    catch (Exception ex)
    {
        return -1;
    }
}

```

```

public async Task<Product> UpdateProduct(Product product)
{
    try
    {
        using (context)
        {
            Product currProduct = context.Products.Find(product.ID);

```

```

        currProduct.Name = product.Name;
        currProduct.Description = product.Description;
        currProduct.ProductCode = product.ProductCode;
        currProduct.ModelFileName = product.ModelFileName;
        currProduct.LibCategoryID = product.LibCategoryID;
        currProduct.CustomCategoryID = product.CustomCategoryID;
        currProduct.Images = product.Images;
        context.Products.Update(currProduct);
        int res = await context.SaveChangesAsync();
        return context.Products.Find(product.ID);
    }
}
catch (Exception ex)
{
    throw ex;
}
}

```

```

public async Task<int> RestoreProduct(int productId)
{
    try
    {
        using (context)
        {
            Product currProduct = context.Products.Find(productId);
            currProduct.Status = Status.Active;
            int res = await context.SaveChangesAsync();
            return res;
        }
    }
    catch (Exception ex)
    {
        return -1;
    }
}

```

```

public Task<Company> UpdateCompany(UpdateCompanyModel company)
{

```

```

        throw new NotImplementedException();
    }

    public async Task<int> RestoreCompany(int companyId)
    {
        try
        {
            using (context)
            {
                Company currCompany = context.Companies.Find(companyId);
                currCompany.Status = Status.Active;
                var companyProducts = context.Products.Where(x => x.CompanyID == companyId);
                foreach (Product product in companyProducts)
                {
                    product.Status = Status.Blocked;
                }
                int res = await context.SaveChangesAsync();
                return res;
            }
        }
        catch (Exception ex)
        {
            return -1;
        }
    }

    public async Task<List<CompanySetting>> GetCompanySettings(int companyId)
    {
        try
        {
            using (context)
            {
                var res = await (from setting in context.CompanySettings
                                where setting.CompanyID == companyId
                                select setting).ToListAsync();
                return res;
            }
        }
    }

```



```

        catch (Exception ex)
        {
            return new List<CompanySetting>();
        }
    }

    public async Task<int> UpdateCompanySettings(CompanySetting setting)
    {
        try
        {
            using (context)
            {
                CompanySetting currSetting = context.CompanySettings.Find(setting.ID);
                currSetting.Value = setting.Value;
                int res = await context.SaveChangesAsync();
                return res;
            }
        }
        catch (Exception ex)
        {
            return -1;
        }
    }

    public async Task<Contact> GetProductContact(int productId)
    {
        try
        {
            using (context)
            {
                var res = await(from contact in context.Contacts
                                where contact.ProductID == productId
                                select contact).FirstOrDefaultAsync();
                return res;
            }
        }
        catch (Exception ex)
        {

```

```

        return null;
    }
}

public async Task<int> UpdateProductContact(Contact contact)
{
    try
    {
        using (context)
        {
            Contact currContact = context.Contacts.Find(contact.ID);
            currContact.Email = contact.Email;
            currContact.Instagram = contact.Instagram;
            currContact.Phone = contact.Phone;
            currContact.ProductLink = contact.ProductLink;
            currContact.Telegram = contact.Telegram;
            currContact.Viber = contact.Viber;
            int res = await context.SaveChangesAsync();
            return res;
        }
    }
    catch (Exception ex)
    {
        return -1;
    }
}
}

```